



# IJCNLP2013

THE 6TH INTERNATIONAL JOINT CONFERENCE ON  
NATURAL LANGUAGE PROCESSING

OCTOBER 14-18, 2013

NAGOYA CONGRESS CENTER, NAGOYA, JAPAN

# PROCEEDINGS

Sixth International Joint Conference on  
Natural Language Processing



**Proceedings of the Main Conference**

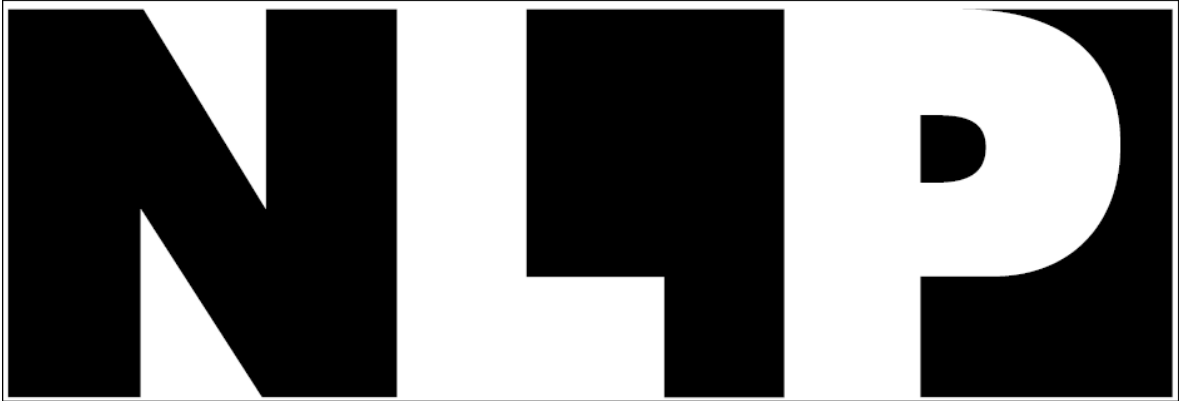




We wish to thank our sponsors and supporters!

Platinum Sponsors

---



[www.anlp.jp](http://www.anlp.jp)

Silver Sponsors

---



[www.google.com](http://www.google.com)

Bronze Sponsors

---



[www.rakuten.com](http://www.rakuten.com)

Supporters

---



**NAGOYA CONVENTION  
& VISITORS BUREAU**

[Nagoya Convention & Visitors Bureau](http://Nagoya Convention & Visitors Bureau)

We wish to thank our organizers!

Organizers

---



[Asian Federation of Natural Language Processing \(AFNLP\)](#)



[Toyohashi University of Technology](#)

©2013 Asian Federation of Natural Language Processing

ISBN 978-4-9907348-0-0

## FOREWORD

Welcome to the 6th International Joint Conference on Natural Language Processing (IJCNLP) in Nagoya, Japan. IJCNLP was initiated in 2004 by The Asian Federation of Natural Language Processing (AFNLP) with the major goal to provide a platform for researchers and professionals from around the world to share their experiences related to natural language processing and computational linguistics. In the past years, IJCNLPs were held in 5 different places: Hainan Island (2004), Jeju Island (2005), Hyderabad (2008), Singapore (2009), and Chiang Mai (2011). This year the 6th IJCNLP is held in Nagoya Congress Center on October 14-18, 2013. The conference covers a broad spectrum of technical areas related to natural language and computation. Besides main conference, the program includes 2 keynote speeches, 3 tutorials, 12 demonstrations, and 7 workshops.

We would like to thank many people who contribute their efforts to IJCNLP 2013. Program chairs Ruslan Mitkov and Jong C. Park select a strong set of papers and organize a wonderful program. PC coordinators Jin-Woo Chung and Isabel Duran support authors and PC committee a stable submission and review platform. Workshop chairs Naoaki Okazaki and Scott Wen-tau Yih organize 7 nice pre-conference and post-conference workshops. Tutorial chairs Vincent Ng and Satoshi Sekine choose 3 very good tutorials. Demo chairs Hang Li and Kentaro Torisawa recommend 12 demonstrations. Sponsorship chair Hiromi Nakaiwa designs sponsor packages and finds financial supports. We thank all the sponsors. Publicity chairs Gareth Jones, Gary Geunbae Lee, Diego Mollá-Aliod, Chengqing Zong and Stajner Sanja help circulate the conference information and promote the conference. We would like to express our special thanks to publication chairs Jing Jiang and Lun-Wei Ku. They bore two babies during the organization of the conference. After the hard work, they deliver an excellent proceeding to the participants. Finally, we are very thankful to those people who dedicate their time and energy to IJCNLP 2013, but are not mentioned in the above. Without them, we would not have had a successful conference.

Hsin-Hsi Chen, General Chair, National Taiwan University, Taiwan  
Hitoshi Isahara, Organization Chair, Toyohashi University of Technology, Japan

October 15, 2013

## PREFACE

As the flagship conference of the Asian Federation of Natural Language Processing (AFNLP), IJCNLP continues to establish itself as a highly influential international event. IJCNLP 2013 covers a broad spectrum of technical areas related to Natural Language Processing. The conference includes regular papers, short papers, poster papers, and system demonstrations, as well as pre- and post-conference tutorials and workshops.

This year, we received 363 paper submissions, which is not as many as the record submissions in the previous conference (e.g., 478 in 2011) but still quite admirable, considering the tough competition for good papers this year, with EMNLP 2013 and RANLP 2013, and the fact that many organizations have restricted their spending in the current economic climate. This represents increasing interest in research on NLP and the growing reputation of IJCNLP as an international event. The 363 submissions include 235 regular, 86 short, and 42 poster paper submissions from more than 37 countries. In particular, approximately 63% of the papers are from 14 countries and areas in Asia Pacific, 18% from 14 countries in Europe, 14% from the United States and Canada; in addition, 4% of the papers are from the Middle East and Africa, and 1% come from South America.

We would like to thank all the authors for submitting papers to IJCNLP 2013. The significant increase in the number of submissions, the topics covered and the wide range of demographic areas represent a rapid and steady growth of our field and hold promise for a bright future. We would also like to thank the 23 area chairs and 439 program committee members for writing over 1078 reviews and meta-reviews and for paving the way for the final paper selection. Of all 363 submissions, a total of 88 papers were accepted as regular papers, representing a healthy 24.4% acceptance rate. Additional 56 papers were accepted as short papers, which, together with regular papers, represent a 39.8% acceptance rate. In addition, 74 papers were accepted as poster papers. Due to various reasons, some authors of accepted papers chose to withdraw their submissions afterwards. As a result, we have 85 regular papers (23.4% acceptance rate), 53 short papers (38.0% acceptance rate), and 62 poster papers. All the regular and short papers are presented orally, and all the poster papers are presented in the plenary poster session. We are extremely grateful to the area chairs and program committee members for all their hard work, without which the preparation of this program would not have been possible. The help of PC coordinators is also much appreciated.

We are delighted to have two keynote speakers addressing different aspects of NLP in IJCNLP 2013. Hwee Tou Ng will present a talk about improving students' writing with automated grammatical error correction, including the review of recent research and advances in grammatical error correction. Roberto Navigli will present a talk about BabelNet 2.0, a very large multilingual semantic network that covers 50 languages and provides both lexicographic and encyclopedic knowledge for all the open-class parts of speech. These plenary talks will surely be not only informative but also enlightening to the audience, leading to many innovative research ideas. We would like to thank General Chair Hsin-Hsi Chen, the Local Arrangements Committee headed by Hitoshi Isahara, and the AFNLP Conference Coordination Committee chaired by Yuji Matsumoto, for their help and advice. Thanks to Jing Jiang and Lun-Wei Ku, the Publication Committee Chairs, for putting the proceedings together, and all the other committee chairs for their great work.

We hope that you enjoy the conference!

Ruslan Mitkov, University of Wolverhampton, England, United Kingdom  
Jong C. Park, Korea Advanced Institute of Science and Technology, Republic of Korea  
IJCNLP 2013 Program Committee Chairs

October 15, 2013



**General Chair**

Hsin-Hsi Chen, National Taiwan University, Taiwan

**Program Committee Chairs**

Ruslan Mitkov, University of Wolverhampton, UK  
Jong C. Park, KAIST, Korea

**Local Organization Committee Chair**

Hitoshi Isahara, Toyohashi University of Technology, Japan

**Workshop Committee Chairs**

Naoaki Okazaki, Tohoku University, Japan  
Scott Wen-tau Yih, Microsoft Research, USA

**Tutorial Chairs**

Vincent Ng, The University of Texas at Dallas, USA  
Satoshi Sekine, New York University, USA

**Demo Chairs**

Hang Li, Huawei Technologies Co., China  
Kentaro Torisawa, NICT, Japan

**Sponsorship Committee Chair**

Hiromi Nakaiwa, NTT, Japan

**Publication Committee Chairs**

Jing Jiang, Singapore Management University, Singapore  
Lun-Wei Ku, Academia Sinica, Taiwan

**Finance Committee Chairs**

Masayuki Okabe, Toyohashi University of Technology, Japan  
Masatoshi Tsuchiya, Toyohashi University of Technology, Japan

**Publicity Committee Chairs**

Gareth Jones, Dublin City University, Ireland  
Gary Geunbae Lee, POSTECH, Korea  
Diego Mollá-Aliod, Macquarie University, Australia  
Chengqing Zong, Chinese Academy of Sciences, China

**PC Coodinators**



Jin-Woo Chung, KAIST, Korea  
Isabel Duran, University of Wolverhampton, UK

## **Area Chairs**

### **Phonology and Morphology**

Mans Hulden, University of Arizona, USA

### **Syntax and Semantics**

Mary Dalrymple, University of Oxford, UK

### **Pragmatics and Discourse**

Joey Frazee, University of Texas, USA

### **Dialogue and Dialogue Systems**

Gary Geunbae Lee, POSTECH, South Korea

### **Language Resources**

Key-Sun Choi, KAIST, South Korea

Doaa Samy, Cairo University, Egypt

### **Statistical and ML Language Models**

Fumiyo Fukumoto, University of Yamanashi, Japan

Leonor Becerra Bonache, Universitat Rovira i Virgili, Spain

### **POS Tagging and Parsing**

Sandra Kuebler, Indiana University, USA

Yusuke Miyao, National Institute of Informatics, Japan

### **Semantic Processing**

Alessandro Moschitti, University of Trento, Italy

Idan Szpektor, Yahoo! Research

### **Information Extraction**

Nigel Collier, National Institute of Informatics, Japan

Jin-Dong Kim, Database Center for Life Science, Japan

### **Text Summarisation**

Inderjeet Mani, Yahoo! Labs, Sunnyvale, USA

Helen Meng, Chinese University of Hong Kong, Hong Kong

### **Information Retrieval and QA**

Qiaozhu Mei, University of Michigan, USA

Iustin Dornescu University of Wolverhampton, UK

### **Text Mining**

Wai Lam, Chinese University of Hong Kong, Hong Kong

## **Opinion Mining**

Alfonso Urena, University of Jaen, Spain

## **NLP for Educational Applications**

Jin-Dong Kim, Database Center for Life Science, Japan

Nigel Collier, National Institute of Informatics, Japan

## **Recent NLP Applications**

Constantin Orasan, University of Wolverhampton, UK

## **Machine Translation**

Dekai Wu, The Hong Kong University of Science and Technology, Hong Kong

Young-suk Lee, IBM, USA

## **Reviewers**

Muhammad Abdul-Mageed, Amjad Abu-Jbara, Hani AbuSalem, Karteek Addanki, Khurshid Ahmad, Akiko Aizawa, Ahmet Aker, Iñaki Alegria, Hadi Amiri, Dana Angluin, Kenji Araki, Eiji Aramaki, Ash Asudeh, Sören Auer, Nguyen Bach, Amit Bagga, Alexandra Balahur, Rafael E. Banchs, Yang Bao, Roberto Basili, Cosmin Bejan, Gemma Bel Enguix, Núria Bel, Lidong Bing, Phil Blunsom, Igor Boguslavsky, Bernd Bohnet, Ilaria Bordino, Stefano Borgo, Johan Bos, Houda Bouamor, Chris Brew, Christopher Brewster, Christopher Brown, Miriam Butt, Aoife Cahill, Lynne Cahill, Nicoletta Calzolari, Erik Cambria, Burcu Can, Marie Candito, Marine Carpuat, Fabio Celli, Daniel Cer, Hutchatai Chanlekha, Wanxiang Che, Berlin Chen, Boxing Chen, Wenliang Chen, Ying Chen, Pu-Jen Cheng, Colin Cherry, David Chiang, Han-Cheol Cho, Jinho D. Choi, Miranda Chong, Khalid Choukri, Janara Christensen, Grzegorz Chrupała, Kevin Cohen, Trevor Cohn, Nigel Collier, Kevyn Collins-Thompson, Gao Cong, John Conroy, Mike Conway, Justin Cope, Bonaventura Coppola, Anna Corazza, Josep Maria Crego, Dan Cristea, Danilo Croce, Andras Csomai, Iria da Cunha, Kareem Darwish, Amitava Das, Adrià de Gispert, Gerard de Melo, Marie-Catherine de Marneffe, Thierry Declerck, Adrian Horia Dediu, Rodolfo Delmonte, Dina Demner-Fushman, Barbara Di Eugenio, Giorgio Maria Di Nunzio, Mona Diab, Markus Dickinson, Marco Dinarelli, Liviu Dinu, Quang Do, Son Doan, Isabel Durán-Muñoz, Ismail El Maarouf, Michael Elhadad, Ossama Emam, Tomaž Erjavec, Gülşen Eryiğit, Maxine Eskenasi, Andrea Esuli, Richard Evans, James Fan, Richárd Farkas, Óscar Ferrández, Corina Forascu, Jennifer Foster, Wei Gao, Claire Gardent, Anna Lisa Gentile, Matthew Gerber, Alec Go, Koldo Gojenola Gallettebeitia, José Miguel Goñi-Menoyo, Edward Grefenstette, Jiafeng Guo, Iryna Gurevych, Francisco Guzman, Amaury Habrard, Keith Hall, Olivier Hamon, Dan Han, Xianpei Han, Laura Hasler, Hany Hassan, Jiyin He, Yulan He, Jeffrey Heinz, James Henderson, Sanjika Hewavitharana, Tsutomu Hirao, Hieu Hoang, Matthew Honnibal, Veronique Hoste, Dirk Hovy, Fei Huang, Zhongqiang Huang, Young-Sook Hwang, Nancy Ide, Adrian Iftene, Radu Ion, Hitoshi Isahara, Guillaume Jacquet, Minwoo Jeong, Sittichai Jiampojarn, Wenbin Jiang, Yunliang Jiang, Maria Dolores Jimenez Lopez, Richard Johansson, Kristiina Jokinen, Senay Kafkas, Min-Yen Kan, Nattiya Kanhabua, Ron Kaplan, Makoto Kato, Mitesh M. Khapra, Adam Kilgarriff, Byeongchang Kim, Hong Kook Kim, Jee-Hyub Kim, Jung-Jae Kim, Seokhwan Kim, Soo-Min Kim, Sungchul Kim, Sun Kim, Woosung Kim, Tracy Holloway King, Philipp Koehn, Oskar Kohonen, Oleksandr Kolomiyets, Mamoru Komachi, Natalia Konstantinova, Hahn Koo, Valia Kordoni, Ioannis Korkontzelos, Alexander Kotov, Zornitsa Kozareva, Udo Kruschwitz, Jonas Kuhn, Sadao Kurohashi, Sobha Lalitha Devi, Wai Lam, Guy Lapalme, Jey Han Lau, Raymond Lau, Alberto Lavelli, Ho-Joon Lee, Jae Sung Lee, Sungjin Lee, Young-Suk Lee, Alessandro Lenci, Yves LePage, Gina-Anne Levow, Si Li, Maria Liakata, Chin-Yew Lin, Christina Lioma, Bing Liu, Haibin Liu, Kang Liu, Qiaoling Liu, Qun

Liu, Yang Liu, Yiqun Liu, Zhiyuan Liu, Avishay Livne, Elena Lloret, Chi-kiu Lo, Christoph Lofi, Oier Lopez de Lacalle, Ramon Lopez-Cozar, Bin Lu, Yue Lu, Zhiyong Lu, Giorgio Magri, Wolfgang Maier, Suresh Manandhar, Inderjeet Mani, Maite Martin, David Martinez, Paloma Martínez, Patricio Martinez-Barco, Eugenio Martínez-Cámara, Yuji Matsumoto, Yutaka Matsuo, Suguru Matsuyoshi, Takuya Matsuzaki, Evgeny Matusov, Arne Mauser, Mark Maybury, Diana McCarthy, John Philip McCrae, Beata Megyesi, Yashar Mehdad, Edgar Meij, Wolfgang Menzel, Farid Meziane, Haitao Mi, Bonan Min, Hye-Jin Min, Wolfgang Minker, Shachar Mirkin, Teruhisa Misu, Makoto Miwa, Yusuke Miyao, Manuel Montes, Alessandro Moschitti, Arjun Mukherjee, Rafael Muñoz-Guillena, Louise Mycock, Satoshi Nakamura, Preslav Nakov, Jason Naradowsky, Vivi Nastase, Costanza Navarretta, Roberto Navigli, Jun Ping Ng, Vincent Ng, ThuyLinh Nguyen, Truc-Vien T. Nguyen, Hitoshi Nishikawa, Joakim Nivre, Josef Novak, Kemal Oflazer, Jong-Hoon Oh, Kousaku Okubo, Petya Osenova, Alexander Osherenko, Shiyang Ou, Arzucan Özgür, Georgios Paltoglou, Rebecca J. Passonneau, Jong Park, Jungyeul Park, Siddharth Patwardhan, Michael Paul, Ted Pedersen, Viktor Pekar, Jose Manuel Perea-Ortega, Wim Peters, Michael Piotrowski, Emily Pitler, Barbara Plank, Natalia Ponomareva, Simone Paolo Ponzetto, Maja Popović, Christopher Potts, Sameer Pradhan, John Prager, Sampo Pyysalo, Vahed Qazvinian, Yao Qian, Tao Qin, Sri-ran Raghavan, Ganesh Ramakrishnan, Owen Rambow, Ines Rehbein, Luz Rello, German Rigau, Fabio Rinaldi, Horacio Rodriguez, Laurent Romary, Paolo Rosso, Bogdan Sacaleanu, Markus Saers, Kenji Sagae, Horacio Saggion, Benoît Sagot, Patrick Saint-Dizier, Agnes Sandor, Marina Santini, Diana Santos, Anoop Sarkar, Kenji Satou, Helmut Schmid, Lane Schwartz, Djamé Seddah, Kazuhiro Seki, Satoshi Sekine, Hendra Setiawan, Aliaksei Severyn, Khaled Shaalan, Libin Shen, Wade Shen, Hiroyuki Shindo, Hiroyuki Shinnou, Eyal Shnarch, Khalil Sima'an, Kiril Simov, Anders Søgaard, Edward Stabler, Sanja Štajner, Efstathios Stamatatos, Pontus Stenetorp, Matthew Stone, Michael Strube, Jian Su, Keh-Yih Su, Kazunari Sugiyama, Ang Sun, Yizhou Sun, Jun Suzuki, Yoshimi Suzuki, Stan Szpakowicz, Hiroya Takamura, Chenhao Tan, Liling Tan, Yee Fan Tan, Ivan Titov, Noriko Tomuro, Sara Tonelli, Lamia Tounsi, Harald Trost, Jose A. Troyano, Manos Tsagkias, Richard Tzong-Han Tsai, Yuen-Hsien Tseng, Yoshimasa Tsuruoka, Kateryna Tymoshenko, Mike Unwalla, Olga Uryupina, Takehito Utsuro, Kees van Deemter, Josef van Genabith, Gertjan van Noord, Menno van Zaanen, Adriano Veloso, Yannick Versley, Karin Verspoor, Jose Vicedo, Laure Vieu, Clare Voss, Vinod Vydiswaran, Henning Wachsmuth, Hui Wan, Xiaojun Wan, Haifeng Wang, Hongning Wang, Hsin-Min Wang, Lidan Wang, Yue Wang, Yu Wang, Houfeng Wang, Leo Wanner, Taro Watanabe, Bonnie Webber, William Webber, Deyi Xiong, Peng Xu, Yasunori Yamamoto, Anssi Yli-Jyrä, Naoki Yoshinaga, Heng Yu, Bei Yu, Fabio Massimo Zanzotto, Duo Zhang, Joy Ying Zhang, Lanbo Zhang, Min Zhang, Qi Zhang, Yi Zhang, Yue Zhang, Ziqi Zhang, Bing Zhao, Le Zhao, Tiejun Zhao, Desislava Zhekova, Jing Zheng, Guodong Zhou, Xiaodan Zhu, Heike Zinsmeister, Michael Zock

## Table of Contents

<i>Semi-Supervised Answer Extraction from Discussion Forums</i>	
Rose Catherine, Rashmi Gangadharaiyah, Karthik Visweswariah and Dinesh Raghu .....	1
<i>WordTopic-MultiRank: A New Method for Automatic Keyphrase Extraction</i>	
Fan Zhang, Lian'en Huang and Bo Peng .....	10
<i>Towards Contextual Healthiness Classification of Food Items - A Linguistic Approach</i>	
Michael Wiegand and Dietrich Klakow .....	19
<i>Learning a Replacement Model for Query Segmentation with Consistency in Search Logs</i>	
Wei Zhang, Yunbo Cao, Chin-Yew Lin, Jian Su and Chew-Lim Tan .....	28
<i>Precise Information Retrieval Exploiting Predicate-Argument Structures</i>	
Daisuke Kawahara, Keiji Shinzato, Tomohide Shibata and Sadao Kurohashi .....	37
<i>Global Model for Hierarchical Multi-Label Text Classification</i>	
Yugo Murawaki .....	46
<i>(Pre-)Annotation of Topic-Focus Articulation in Prague Czech-English Dependency Treebank</i>	
Jiří Mírovský, Kateřina Rysová, Magdaléna Rysová and Eva Hajičová .....	55
<i>Animacy Acquisition Using Morphological Case</i>	
Riyaz Ahmad Bhat and Dipti Misra Sharma .....	64
<i>The Complexity of Math Problems – Linguistic, or Computational?</i>	
Takuya Matsuzaki, Hidenao Iwane, Hirokazu Anai and Noriko Arai .....	73
<i>Hybrid Models for Lexical Acquisition of Correlated Styles</i>	
Julian Brooke and Graeme Hirst .....	82
<i>Introducing the Prague Discourse Treebank 1.0</i>	
Lucie Poláková, Jiří Mírovský, Anna Nedoluzhko, Pavlína Jínová, Šárka Zikánová and Eva Hajičová .....	91
<i>Multilingual Mention Detection for Coreference Resolution</i>	
Olga Uryupina and Alessandro Moschitti .....	100
<i>A Weakly Supervised Bayesian Model for Violence Detection in Social Media</i>	
Amparo Elizabeth Cano Basave, Yulan He, Kang Liu and Jun Zhao .....	109
<i>Detecting Spammers in Community Question Answering</i>	
Zhuoye Ding, Yeyun Gong, Yaqian Zhou, Qi Zhang and Xuanjing Huang .....	118
<i>Chinese Informal Word Normalization: an Experimental Study</i>	
Aobo Wang, Min-Yen Kan, Daniel Andrade, Takashi Onishi and Kai Ishikawa .....	127
<i>Feature Selection Using a Semantic Hierarchy for Event Recognition and Type Classification</i>	
Yoonjae Jeong and Sung-Hyon Myaeng .....	136
<i>Romanization-based Approach to Morphological Analysis in Korean SMS Text Processing</i>	
Youngsam Kim and Hyopil Shin .....	145

<i>Efficient Word Lattice Generation for Joint Word Segmentation and POS Tagging in Japanese</i> Nobuhiro Kaji and Masaru Kitsuregawa.....	153
<i>A Simple Approach to Unknown Word Processing in Japanese Morphological Analysis</i> Ryohei Sasano, Sadao Kurohashi and Manabu Okumura.....	162
<i>Chinese Word Segmentation by Mining Maximized Substrings</i> Mo Shen, Daisuke Kawahara and Sadao Kurohashi .....	171
<i>Capturing Long-distance Dependencies in Sequence Models: A Case Study of Chinese Part-of-speech Tagging</i> Weiwei Sun, Xiaochang Peng and Xiaojun Wan .....	180
<i>Exploring Semantic Information in Hindi WordNet for Hindi Dependency Parsing</i> Sambhav Jain, Naman Jain, Aniruddha Tammewar, Riyaz Ahmad Bhat and Dipti Sharma.....	189
<i>Towards Robust Cross-Domain Domain Adaptation for Part-of-Speech Tagging</i> Tobias Schnabel and Hinrich Schütze .....	198
<i>Dependency Parsing for Identifying Hungarian Light Verb Constructions</i> Veronika Vincze, János Zsibrita and István Nagy T. ....	207
<i>Written Dialog and Social Power: Manifestations of Different Types of Power in Dialog Behavior</i> Vinodkumar Prabhakaran and Owen Rambow .....	216
<i>Evaluation of the Scusi? Spoken Language Interpretation System – A Case Study</i> Thomas Kleinbauer, Ingrid Zukerman and Su Nam Kim .....	225
<i>A Noisy Channel Approach to Error Correction in Spoken Referring Expressions</i> Su Nam Kim, Ingrid Zukerman, Thomas Kleinbauer and Farshid Zavareh .....	234
<i>Natural Language Query Refinement for Problem Resolution from Crowd-Sourced Semi-Structured Data</i> Rashmi Gangadharaiah and Balakrishnan Narayanaswamy.....	243
<i>Ensemble Triangulation for Statistical Machine Translation</i> Majid Razmara and Anoop Sarkar.....	252
<i>Robust Transliteration Mining from Comparable Corpora with Bilingual Topic Models</i> John Richardson, Toshiaki Nakazawa and Sadao Kurohashi .....	261
<i>SuMT: A Framework of Summarization and MT</i> Houda Bouamor, Behrang Mohit and Kemal Oflazer .....	270
<i>Tuning SMT with a Large Number of Features via Online Feature Grouping</i> Lemao Liu, Tiejun Zhao, Taro Watanabe and Eiichiro Sumita .....	279
<i>Multimodal Comparable Corpora as Resources for Extracting Parallel Data: Parallel Phrases Extraction</i> Haithem Afli, Loïc Barrault and Holger Schwenk .....	286
<i>Bootstrapping Large-scale Named Entities using URL-Text Hybrid Patterns</i> Chao Zhang, Shiqi Zhao and Haifeng Wang .....	293
<i>Feature-Rich Segment-Based News Event Detection on Twitter</i> Yanxia Qin, Yue Zhang, Min Zhang and Dequan Zheng .....	302

<i>Building Chinese Event Type Paradigm Based on Trigger Clustering</i> Xiao Ding, Bing Qin and Ting Liu .....	311
<i>Chinese Named Entity Abbreviation Generation Using First-Order Logic</i> Huan Chen, Qi Zhang, Jin Qian and Xuanjing Huang .....	320
<i>Full-coverage Identification of English Light Verb Constructions</i> István Nagy T., Veronika Vincze and Richárd Farkas .....	329
<i>Detecting Deceptive Opinions with Profile Compatibility</i> Vanessa Wei Feng and Graeme Hirst .....	338
<i>Behind the Times: Detecting Epoch Changes using Large Corpora</i> Octavian Popescu and Carlo Strapparava .....	347
<i>How Noisy Social Media Text, How Different Social Media Sources?</i> Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay and Li Wang .....	356
<i>Who Had the Upper Hand? Ranking Participants of Interactions Based on Their Relative Power</i> Vinodkumar Prabhakaran, Ajita John and Dorée D. Seligmann .....	365
<i>Readability Indices for Automatic Evaluation of Text Simplification Systems: A Feasibility Study for Spanish</i> Sanja Štajner and Horacio Saggion .....	374
<i>Weasels, Hedges and Peacocks: Discourse-level Uncertainty in Wikipedia Articles</i> Veronika Vincze .....	383
<i>Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia</i> Fahd Alotaibi and Mark Lee .....	392
<i>Ranking Translation Candidates Acquired from Comparable Corpora</i> Rima Harastani, Béatrice Daille and Emmanuel Morin .....	401
<i>Using the Semantic-Syntactic Interface for Reliable Arabic Modality Annotation</i> Rania Al-Sabbagh, Jana Diesner and Roxana Girju .....	410
<i>Mapping Rules for Building a Tunisian Dialect Lexicon and Generating Corpora</i> Rahma Boujelbane, Mariem Ellouze Khemekhem and Lamia Hadrach Belguith .....	419
<i>Hypothesis Refinement Using Agreement Constraints in Machine Translation</i> Ankur Gandhe and Rashmi Gangadharaiah .....	429
<i>Scalable Variational Inference for Extracting Hierarchical Phrase-based Translation Rules</i> Baskaran Sankaran, Gholamreza Haffari and Anoop Sarkar .....	438
<i>A Topic-Triggered Language Model for Statistical Machine Translation</i> Heng Yu, Jinsong Su, Yajuan Lv and Qun Liu .....	447
<i>Reserved Self-training: A Semi-supervised Sentiment Classification Method for Chinese Microblogs</i> Zhiguang Liu, Xishuang Dong, Yi Guan and Jinfeng Yang .....	455
<i>Enhancing Lexicon-Based Review Classification by Merging and Revising Sentiment Dictionaries</i> Heeryon Cho, Jong-Seok Lee and Songkuk Kim .....	463

<i>Exploring the Effects of Word Roots for Arabic Sentiment Analysis</i> Shereen Oraby, Yasser El-Sonbaty and Mohamad Abou El-Nasr .....	471
<i>Topical Key Concept Extraction from Folksonomy</i> Han Xue, Bing Qin, Ting Liu and Chao Xiang .....	480
<i>Uncovering Distributional Differences between Synonyms and Antonyms in a Word Space Model</i> Silke Scheible, Sabine Schulte im Walde and Sylvia Springorum .....	489
<i>Multilingual Word Sense Disambiguation Using Wikipedia</i> Bharath Dandala, Rada Mihalcea and Razvan Bunescu .....	498
<i>Semantic v.s. Positions: Utilizing Balanced Proximity in Language Model Smoothing for Information Retrieval</i> Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv and Xiaoming Li .....	507
<i>An Unsupervised Parameter Estimation Algorithm for a Generative Dependency N-gram Language Model</i> Chenchen Ding and Mikio Yamamoto .....	516
<i>Learning a Product of Experts with Elitist Lasso</i> Mengqiu Wang and Christopher D. Manning .....	525
<i>Learning Efficient Information Extraction on Heterogeneous Texts</i> Henning Wachsmuth, Benno Stein and Gregor Engels .....	534
<i>TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction</i> Adrien Bougouin, Florian Boudin and Béatrice Daille .....	543
<i>Understanding the Semantic Intent of Natural Language Query</i> Juan Xu, Qi Zhang and Xuanjing Huang .....	552
<i>Sentiment Classification for Movie Reviews in Chinese Using Parsing-based Methods</i> Wen-Juan Hou and Chuang-Ping Chang .....	561
<i>Sentiment Aggregation using ConceptNet Ontology</i> Subhabrata Mukherjee and Sachindra Joshi .....	570
<i>Detecting Cyberbullying Entries on Informal School Websites Based on Category Relevance Maximization</i> Taisei Nitta, Fumito Masui, Michal Ptaszynski, Yasutomo Kimura, Rafal Rzepka and Kenji Araki	579
<i>A Lexicon-based Investigation of Research Issues in Japanese Factuality Analysis</i> Kazuya Narita, Junta Mizuno and Kentaro Inui .....	587
<i>A Hierarchical Semantics-Aware Distributional Similarity Scheme</i> Shuqi Sun, Ke Sun, Shiqi Zhao, Haifeng Wang, Muyun Yang and Sheng Li .....	596
<i>Labeled Alignment for Recognizing Textual Entailment</i> Xiaolin Wang, Hai Zhao and Bao-Liang Lu .....	605
<i>Context-Based Chinese Word Segmentation using SVM Machine-Learning Algorithm without Dictionary Support</i> Chia-ming Lee and Chien-Kang Huang .....	614

<i>A Common Case of Jekyll and Hyde: The Synergistic Effect of Using Divided Source Training Data for Feature Augmentation</i>	
Yan Song and Fei Xia .....	623
<i>Detecting Polysemy in Hard and Soft Cluster Analyses of German Preposition Vector Spaces</i>	
Sylvia Springorum, Sabine Schulte im Walde and Jason Utt .....	632
<i>Generalized Abbreviation Prediction with Negative Full Forms and Its Application on Improving Chinese Web Search</i>	
Xu Sun, Wenjie Li, Fanqi Meng and Houfeng Wang .....	641
<i>Prosody-Based Unsupervised Speech Summarization with Two-Layer Mutually Reinforced Random Walk</i>	
Sujay Kumar Jauhar, Yun-Nung Chen and Florian Metze .....	648
<i>Mining the Gaps: Towards Polynomial Summarization</i>	
Marina Litvak and Natalia Vanetik .....	655
<i>Detecting Domain Dedicated Polar Words</i>	
Raksha Sharma and Pushpak Bhattacharyya .....	661
<i>Can I Hear You? Sentiment Analysis on Medical Forums</i>	
Tanveer Ali, David Schramm, Marina Sokolova and Diana Inkpen .....	667
<i>Construction of Emotional Lexicon Using Potts Model</i>	
Braja Gopal Patra, Hiroya Takamura, Dipankar Das, Manabu Okumura and Sivaji Bandyopadhyay	674
<i>Suicidal Tendencies: The Automatic Classification of Suicidal and Non-Suicidal Lyricists Using NLP</i>	
Matthew Mulholland and Joanne Quinn .....	680
<i>Unsupervised Word Class Induction for Under-resourced Languages: A Case Study on Indonesian</i>	
Meladel Mistica, Jey Han Lau and Timothy Baldwin .....	685
<i>An Efficient Active Learning Framework for New Relation Types</i>	
Lisheng Fu and Ralph Grishman .....	692
<i>Parsing Dependency Paths to Identify Event-Argument Relations</i>	
Seung-Cheol Baek and Jong Park .....	699
<i>Augmentable Paraphrase Extraction Framework</i>	
MeiHua Chen, YiChun Chen, ShihTing Huang and Jason S. Chang .....	706
<i>Automatic Prediction of Evidence-based Recommendations via Sentence-level Polarity Classification</i>	
Abeed Sarker, Diego Mollá-Aliod and Cécile Paris .....	712
<i>Clustering Microtext Streams for Event Identification</i>	
Jie Yin .....	719
<i>Automatic Corpora Construction for Text Classification</i>	
Dandan Wang, Qingcai Chen, Xiaolong Wang and Bingyang Yu .....	726
<i>Learning to Generate Diversified Query Interpretations using Biconvex Optimization</i>	
Ramakrishna Bairi, Ambha A and Ganesh Ramakrishnan .....	733



<i>Learning Based Approaches for Vietnamese Question Classification Using Keywords Extraction from the Web</i>	
Dang Tran, Cuong Chu, Son Pham and Minh Nguyen .....	740
<i>Detecting Bot-Answerable Questions in Ubuntu Chat</i>	
David Uthus and David Aha .....	747
<i>Alignment-based Annotation of Proofreading Texts toward Professional Writing Assistance</i>	
Ngan Nguyen and Yusuke Miyao .....	753
<i>Toward Automatic Processing of English Metalanguage</i>	
Shomir Wilson .....	760
<i>On the Effectiveness of Using Syntactic and Shallow Semantic Tree Kernels for Automatic Assessment of Essays</i>	
Yllias Chali and Sadid A. Hasan .....	767
<i>Little by Little: Semi Supervised Stemming through Stem Set Minimization</i>	
Vasudevan N and Pushpak Bhattacharyya .....	774
<i>What Information is Helpful for Dependency Based Semantic Role Labeling</i>	
Yanyan Luo, Kevin Duh and Yuji Matsumoto .....	781
<i>Classifying Taxonomic Relations between Pairs of Wikipedia Articles</i>	
Or Biran and Kathleen McKeown .....	788
<i>A Rule System for Chinese Time Entity Recognition by Comprehensive Linguistic Study</i>	
Hongzhi Xu and Chu-Ren Huang .....	795
<i>Financial Sentiment Analysis for Risk Prediction</i>	
Chuan-Ju Wang, Ming-Feng Tsai, Tse Liu and Chin-Ting Chang .....	802
<i>Sense Disambiguation: From Natural Language Words to Mathematical Terms</i>	
Minh-Quoc Nghiem, Giovanni Yoko Kristianto, Goran Topic and Akiko Aizawa .....	809
<i>Adapting a State-of-the-art Anaphora Resolution System for Resource-poor Language</i>	
Utpal Sikdar, Asif Ekbal, Sriparna Saha, Olga Uryupina and Massimo Poesio .....	815
<i>Chinese Event Coreference Resolution: Understanding the State of the Art</i>	
Chen Chen and Vincent Ng .....	822
<i>A Two-Step Named Entity Recognizer for Open-Domain Search Queries</i>	
Andreas Eiselt and Alejandro Figueroa .....	829
<i>A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction</i>	
Florian Boudin .....	834
<i>Translating Chinese Unknown Words by Automatically Acquired Templates</i>	
Ming-Hong Bai, Yu-Ming Hsieh, Keh-Jiann Chen and Jason S. Chang .....	839
<i>Multilingual Lexicon Bootstrapping - Improving a Lexicon Induction System Using a Parallel Corpus</i>	
Patrick Ziering, Lonneke van der Plas and Hinrich Schütze .....	844
<i>Mining Japanese Compound Words and Their Pronunciations from Web Pages and Tweets</i>	
Xianchao Wu .....	849

<i>A Factoid Question Answering System Using Answer Pattern Matching</i> Nagehan Pala Er and Ilyas Cicekli .....	854
<i>Chinese Short Text Classification Based on Domain Knowledge</i> Xiao Feng, Yang Shen, Chengyong Liu, Wei Liang and Shuwu Zhang .....	859
<i>Applying Graph-based Keyword Extraction to Document Retrieval</i> Youngsam Kim, Munhyong Kim, Andrew Cattle, Julia Otmakhova, Suzi Park and Hyopil Shin	864
<i>Semi-supervised Classification of Twitter Messages for Organization Name Disambiguation</i> Shu Zhang, Jianwei Wu, Dequan Zheng, Yao Meng and Hao Yu .....	869
<i>Word in a Dictionary is used by Numerous Users</i> Eiji Aramaki, Sachiko Maskawa, Mai Miyabe, Mizuki Morita and Sachi Yasuda .....	874
<i>Extracting Evaluative Conditions from Online Reviews: Toward Enhancing Opinion Mining</i> Yuki Nakayama and Atsushi Fujii .....	878
<i>Cognate Production using Character-based Machine Translation</i> Lisa Beinborn, Torsten Zesch and Iryna Gurevych .....	883
<i>An Empirical Study of Combing Multiple Models in Bengali Question Classification</i> Somnath Banerjee and Sivaji Bandyopadhyay .....	892
<i>A Two-Stage Classifier for Sentiment Analysis</i> Dai Quoc Nguyen, Dat Quoc Nguyen and Son Bao Pham .....	897
<i>Exploiting User Search Sessions for the Semantic Categorization of Question-like Informational Search Queries</i> Alejandro Figueroa and Guenter Neumann .....	902
<i>Influence of Part-of-Speech and Phrasal Category Universal Tag-set in Tree-to-Tree Translation Models</i> Francisco Oliveira, Derek F. Wong, Lidia S. Chao, Liang Tian and Liangye He .....	907
<i>Interest Analysis using PageRank and Social Interaction Content</i> Chung-chi Huang and Lun-Wei Ku .....	912
<i>Time Series Topic Modeling and Bursty Topic Detection of Correlated News and Twitter</i> Daichi Koike, Yusuke Takahashi, Takehito Utsuro, Masaharu Yoshioka and Noriko Kando ....	917
<i>A Distant Supervision Approach for Identifying Perspectives in Unstructured User-Generated Text</i> Attapol Thamrongrattanarit, Colin Pollock, Benjamin Goldenberg and Jason Fennell .....	922
<i>An Approach of Hybrid Hierarchical Structure for Word Similarity Computing by HowNet</i> Jiangming Liu, Jinan Xu and Yujie Zhang .....	927
<i>Extracting Causes of Emotions from Text</i> Alena Neviarouskaya and Masaki Aono .....	932
<i>Automated Grammar Correction Using Hierarchical Phrase-Based Statistical Machine Translation</i> Bibek Behera and Pushpak Bhattacharyya .....	937
<i>Finding Dependency Parsing Limits over a Large Spanish Corpus</i> Muntsa Padró, Miguel Ballesteros, Héctor Martínez and Bernd Bohnet .....	942

<i>High Quality Dependency Selection from Automatic Parses</i> Gongye Jin, Daisuke Kawahara and Sadao Kurohashi .....	947
<i>Building Specialized Bilingual Lexicons Using Word Sense Disambiguation</i> Dhouha Bouamor, Nasredine Semmar and Pierre Zweigenbaum .....	952
<i>Predicate Argument Structure Analysis using Partially Annotated Corpora</i> Koichiro Yoshino, Shinsuke Mori and Tatsuya Kawahara .....	957
<i>Statistical Dialogue Management using Intention Dependency Graph</i> Koichiro Yoshino, Shinji Watanabe, Jonathan Le Roux and John R. Hershey .....	962
<i>Repairing Incorrect Translation with Examples</i> Junguo Zhu, Muyun Yang, Sheng Li and Tiejun Zhao .....	967
<i>Phrase-based Parallel Fragments Extraction from Comparable Corpora</i> Xiaoyin Fu, Wei Wei, Shixiang Lu, Zhenbiao Chen and Bo Xu .....	972
<i>A Hybrid Approach for Anaphora Resolution in Hindi</i> Praveen Dakwale, Vandan Mujadia and Dipti M Sharma .....	977
<i>Structure Cognizant Pseudo Relevance Feedback</i> Arjun Atreya V, Yogesh Kakde, Pushpak Bhattacharyya and Ganesh Ramakrishnan .....	982
<i>Cross-Domain Answer Ranking using Importance Sampling</i> Anders Johannsen and Anders Søgaard .....	987
<i>Morphological Analysis of Tunisian Dialect</i> Inès Zribi, Mariem Ellouze Khemakhem and Lamia Hadrich Belguith .....	992
<i>Disambiguating Explicit Discourse Connectives without Oracles</i> Anders Johannsen and Anders Søgaard .....	997
<i>Updating Rare Term Vector Replacement</i> Tobias Berka and Marian Vajteršic .....	1002
<i>Statistical Morphological Analyzer for Hindi</i> Deepak Kumar Malladi and Prashanth Mannem .....	1007
<i>Induction of Root and Pattern Lexicon for Unsupervised Morphological Analysis of Arabic</i> Bilal Khaliq and John Carrol .....	1012
<i>Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text</i> Minh Quang Nhat Pham, Minh Le Nguyen and Akira Shimazu .....	1017
<i>Using Transliteration of Proper Names from Arabic to Latin Script to Improve English-Arabic Word Alignment</i> Nasredine Semmar and Houda Saadane .....	1022
<i>A Semi-Supervised Method for Arabic Word Sense Disambiguation Using a Weighted Directed Graph</i> Laroussi Merhbene, Anis Zouaghi and Mounir Zrigui .....	1027
<i>Incremental Segmentation and Decoding Strategies for Simultaneous Translation</i> Mahsa Yarmohammadi, Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore and Baskaran Sankaran	1032

<i>Two Case Studies on Translating Pronouns in a Deep Syntax Framework</i> Michal Novák, Zdenek Zabokrtsky and Anna Nedoluzhko .....	1037
<i>Bootstrapping Phrase-based Statistical Machine Translation via WSD Integration</i> Hien Vu Huy, Phuong-Thai Nguyen, Tung-Lam Nguyen and M.L Nguyen .....	1042
<i>Orthographic and Morphological Processing for Persian-to-English Statistical Machine Translation</i> Mohammad Sadegh Rasooli, Ahmed El Kholy and Nizar Habash .....	1047
<i>Interoperability between Service Composition and Processing Pipeline: Case Study on the Language Grid and UIMA</i> Trang Mai Xuan, Yohei Murakami, Donghui Lin and Toru Ishida .....	1052
<i>Improving Calculation of Contextual Similarity for Constructing a Bilingual Dictionary via a Third Language</i> Takashi Tsunakawa, Yosuke Yamamoto and Hiroyuki Kaji .....	1057
<i>Two-Stage Pre-ordering for Japanese-to-English Statistical Machine Translation</i> Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh and Masaaki Nagata .....	1062
<i>Grammatical Error Correction Using Feature Selection and Confidence Tuning</i> Yang Xiang, Yaoyun Zhang, Xiaolong Wang, Chongqiang Wei, Wen Zheng, Xiaoqiang Zhou, Yuxiu Hu and Yang Qin .....	1067
<i>An Online Algorithm for Learning over Constrained Latent Representations using Multiple Views</i> Ann Clifton, Max Whitney and Anoop Sarkar .....	1072
<i>Synonym Acquisition Using Bilingual Comparable Corpora</i> Daniel Andrade, Masaaki Tsuchida, Takashi Onishi and Kai Ishikawa .....	1077
<i>Exploring Verb Frames for Sentence Simplification in Hindi</i> Ankush Soni, Sambhav Jain and Dipti Misra Sharma .....	1082
<i>Dirichlet Processes for Joint Learning of Morphology and PoS Tags</i> Burcu Can and Suresh Manandhar .....	1087
<i>Parser Accuracy in Quality Estimation of Machine Translation: A Tree Kernel Approach</i> Rasoul Samad Zadeh Kaljahi, Jennifer Foster, Raphael Rubino, Johann Roturier and Fred Hollowood .....	1092
<i>Attribute Relation Extraction from Template-inconsistent Semi-structured Text by Leveraging Site-level Knowledge</i> Yang Liu, Fang Liu, Siwei Lai, Kang Liu, Guangyou Zhou and Jun Zhao .....	1097
<i>Optimum Parameter Selection for K.L.D. Based Authorship Attribution in Gujarati</i> Parth Mehta and Prasenjit Majumder .....	1102
<i>Modeling User Leniency and Product Popularity for Sentiment Classification</i> Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji and Masaru Kitsuregawa .....	1107
<i>A Generalized LCS Algorithm and Its Application to Corpus Alignment</i> Jin-Dong Kim .....	1112
<i>Semantic Naïve Bayes Classifier for Document Classification</i> How Jing, Yu Tsao, Kuan-Yu Chen and Hsin-Min Wang .....	1117

<i>Cluster-based Web Summarization</i>	
Yves Petinot, Kathleen McKeown and Kapil Thadani . . . . .	1124
<i>Automated Activity Recognition in Clinical Documents</i>	
Camilo Thorne, Marco Montali, Diego Calvanese, Elena Cardillo and Claudio Eccher . . . . .	1129
<i>Large-Scale Text Collection for Unwritten Languages</i>	
Florian R. Hanke and Steven Bird . . . . .	1134
<i>A Self-learning Template Approach for Recognizing Named Entities from Web Text</i>	
Qian Liu, Bingyang Liu, Dayong Wu, Yue Liu and Xueqi Cheng . . . . .	1139
<i>Accurate Parallel Fragment Extraction from Quasi-Comparable Corpora using Alignment Model and Translation Lexicon</i>	
Chenhui Chu, Toshiaki Nakazawa and Sadao Kurohashi . . . . .	1144
<i>Meta-level Statistical Machine Translation</i>	
Sajad Ebrahimi, Kourosh Meshgi, Shahram Khadivi and Mohammad Ebrahim Shiri Ahmad Abady	
1151	
<i>Bayesian Induction of Bracketing Inversion Transduction Grammars</i>	
Markus Saers and Dekai Wu . . . . .	1158
<i>Estimating the Quality of Translated User-Generated Content</i>	
Raphael Rubino, Jennifer Foster, Rasoul Samad Zadeh Kaljahi, Johann Roturier and Fred Hol-	
lowood . . . . .	1167
<i>Selective Combination of Pivot and Direct Statistical Machine Translation Models</i>	
Ahmed El Kholly, Nizar Habash, Gregor Leusch, Evgeny Matusov and Hassan Sawaf . . . . .	1174
<i>Multiword Expressions in the Context of Statistical Machine Translation</i>	
Mahmoud Ghoneim and Mona Diab . . . . .	1181
<i>Uncertainty Detection for Natural Language Watermarking</i>	
György Szarvas and Iryna Gurevych . . . . .	1188
<i>KySS 1.0: a Framework for Automatic Evaluation of Chinese Input Method Engines</i>	
Zhongye Jia and Hai Zhao . . . . .	1195
<i>Automatic Extraction of Social Networks from Literary Text: A Case Study on Alice in Wonderland</i>	
Apoorv Agarwal, Anup Kotalwar and Owen Rambow . . . . .	1202
<i>Using the Web to Train a Mobile Device Oriented Japanese Input Method Editor</i>	
Xianchao Wu, Rixin Xiao and Xiaoxin Chen . . . . .	1209
<i>A Novel Approach Towards Incorporating Context Processing Capabilities in NLIDB System</i>	
Arjun Akula, Rajeev Sangal and Radhika Mamidi . . . . .	1216
<i>Iterative Development and Evaluation of a Social Conversational Agent</i>	
Annika Silvervarg and Arne Jönsson . . . . .	1223
<i>A Hybrid Morphological Disambiguation System for Turkish</i>	
Mucahid Kutlu and Ilyas Cicekli . . . . .	1230
<i>A Dynamic Confusion Score for Dependency Arc Labels</i>	
Sambhav Jain and Bhasha Agrawal . . . . .	1237

<i>Increasing the Quality and Quantity of Source Language Data for Unsupervised Cross-Lingual POS Tagging</i>	
Long Duong, Paul Cook, Steven Bird and Pavel Pecina . . . . .	1243
<i>Towards the Annotation of Penn TreeBank with Information Structure</i>	
Bernd Bohnet, Alicia Burga and Leo Wanner . . . . .	1250
<i>Constituency and Dependency Relationship from a Tree Adjoining Grammar and Abstract Categorical Grammars Perspective</i>	
Aleksandre Maskharashvili and Sylvain Pogodalla . . . . .	1257
<i>Named Entity Extraction using Information Distance</i>	
Sangameshwar Patil, Sachin Pawar and Girish Palshikar . . . . .	1264
<i>Feature-based Neural Language Model and Chinese Word Segmentation</i>	
Mairgup Mansur, Wenzhe Pei and Baobao Chang . . . . .	1271
<i>Human-Computer Interactive Chinese Word Segmentation: An Adaptive Dirichlet Process Mixture Model Approach</i>	
Tongfei Chen, Xiaojun Zou, Weimeng Zhu and Junfeng Hu . . . . .	1278
<i>Effect of Non-linear Deep Architecture in Sequence Labeling</i>	
Mengqiu Wang and Christopher D. Manning . . . . .	1285
<i>Case Study of Model Adaptation: Transfer Learning and Online Learning</i>	
Kenji Imamura . . . . .	1292
<i>Source and Translation Classification using Most Frequent Words</i>	
Zahurul Islam and Armin Hoenen . . . . .	1299
<i>Comparison of Algorithmic and Human Assessments of Sentence Similarity</i>	
John Mersch and R. Raymond Lang . . . . .	1306
<i>Effective Selectional Restrictions for Unsupervised Relation Extraction</i>	
Alan Akbik, Larysa Visengeriyeva, Johannes Kirschnick and Alexander Löser . . . . .	1312
<i>Bootstrapping Semantic Lexicons for Technical Domains</i>	
Patrick Ziering, Lonneke van der Plas and Hinrich Schütze . . . . .	1321
<i>Long-Distance Time-Event Relation Extraction</i>	
Alessandro Moschitti, Siddharth Patwardhan and Chris Welty . . . . .	1330
<i>Unsupervised Extraction of Attributes and Their Values from Product Description</i>	
Keiji Shinzato and Satoshi Sekine . . . . .	1339
<i>Stance Classification of Ideological Debates: Data, Models, Features, and Constraints</i>	
Kazi Saidul Hasan and Vincent Ng . . . . .	1348
<i>University Entrance Examinations as a Benchmark Resource for NLP-based Problem Solving</i>	
Yusuke Miyao and Ai Kawazoe . . . . .	1357
<i>Linguistically Aware Coreference Evaluation Metrics</i>	
Chen Chen and Vincent Ng . . . . .	1366
<i>An Empirical Assessment of Contemporary Online Media in Ad-Hoc Corpus Creation for Social Events</i>	
Kanika Narang, Seema Nagar, Sameep Mehta, L V Subramaniam and Kuntal Dey . . . . .	1375

<i>Diagnosing Causes of Reading Difficulty using Bayesian Networks</i> Pascual Martínez-Gómez and Akiko Aizawa .....	1383
<i>Word Co-occurrence Counts Prediction for Bilingual Terminology Extraction from Comparable Corpora</i> Amir Hazem and Emmanuel Morin .....	1392
<i>Measuring the Effect of Discourse Relations on Blog Summarization</i> Shamima Mithun and Leila Kosseim .....	1401
<i>Supervised Sentence Fusion with Single-Stage Inference</i> Kapil Thadani and Kathleen McKeown .....	1410
<i>Detecting and Correcting Learner Korean Particle Omission Errors</i> Ross Israel, Markus Dickinson and Sun-Hee Lee .....	1419
<i>Automatic Identification of Learners' Language Background Based on Their Writing in Czech</i> Katsiaryna Aharodnik, Marco Chang, Anna Feldman and Jirka Hana .....	1428

# Conference Program

## October 15, 2013 (Tuesday)

09:00-09:20 Opening (Reception Hall)

09:20-10:20 Keynote Speech - Hwee Tou Ng (National University of Singapore) (Reception Hall)

10:20-10:50 Coffee Break

10:50-12:05 Regular Papers

### Information Extraction I (Reception Hall)

10:50–11:15 *Semi-Supervised Answer Extraction from Discussion Forums*  
Rose Catherine, Rashmi Gangadharaiah, Karthik Visweswariah and Dinesh Raghu

11:15–11:40 *WordTopic-MultiRank: A New Method for Automatic Keyphrase Extraction*  
Fan Zhang, Lian'en Huang and Bo Peng

11:40–12:05 *Towards Contextual Healthiness Classification of Food Items - A Linguistic Approach*  
Michael Wiegand and Dietrich Klakow

### Information Retrieval I (Room 141 + 142)

10:50–11:15 *Learning a Replacement Model for Query Segmentation with Consistency in Search Logs*  
Wei Zhang, Yunbo Cao, Chin-Yew Lin, Jian Su and Chew-Lim Tan

11:15–11:40 *Precise Information Retrieval Exploiting Predicate-Argument Structures*  
Daisuke Kawahara, Keiji Shinzato, Tomohide Shibata and Sadao Kurohashi

11:40–12:05 *Global Model for Hierarchical Multi-Label Text Classification*  
Yugo Murawaki



**October 15, 2013 (Tuesday) (continued)**

**Syntax and Semantics (Room 131 + 132)**

10:50–11:15 *(Pre-)Annotation of Topic-Focus Articulation in Prague Czech-English Dependency Treebank*

Jiří Mírovský, Kateřina Rysová, Magdaléna Rysová and Eva Hajičová

11:15–11:40 *Animacy Acquisition Using Morphological Case*

Riyaz Ahmad Bhat and Dipti Misra Sharma

11:40–12:05 *The Complexity of Math Problems – Linguistic, or Computational?*

Takuya Matsuzaki, Hidenao Iwane, Hirokazu Anai and Noriko Arai

**Pragmatics and Discourse (Room 133 + 134)**

10:50–11:15 *Hybrid Models for Lexical Acquisition of Correlated Styles*

Julian Brooke and Graeme Hirst

11:15–11:40 *Introducing the Prague Discourse Treebank 1.0*

Lucie Poláková, Jiří Mírovský, Anna Nedoluzhko, Pavlína Jínová, Šárka Zikánová and Eva Hajičová

11:40–12:05 *Multilingual Mention Detection for Coreference Resolution*

Olga Uryupina and Alessandro Moschitti

12:05-13:30 Lunch

13:30-15:10 Regular Papers

**October 15, 2013 (Tuesday) (continued)**

**Text Mining (Reception Hall)**

- 13:30–13:55 *A Weakly Supervised Bayesian Model for Violence Detection in Social Media*  
Amparo Elizabeth Cano Basave, Yulan He, Kang Liu and Jun Zhao
- 13:55–14:20 *Detecting Spammers in Community Question Answering*  
Zhuoye Ding, Yeyun Gong, Yaqian Zhou, Qi Zhang and Xuanjing Huang
- 14:20–14:45 *Chinese Informal Word Normalization: an Experimental Study*  
Aobo Wang, Min-Yen Kan, Daniel Andrade, Takashi Onishi and Kai Ishikawa
- 14:45–15:10 *Feature Selection Using a Semantic Hierarchy for Event Recognition and Type Classification*  
Yoonjae Jeong and Sung-Hyon Myaeng

**Phonology and Morphology (Room 141 + 142)**

- 13:30–13:55 *Romanization-based Approach to Morphological Analysis in Korean SMS Text Processing*  
Youngsam Kim and Hyopil Shin
- 13:55–14:20 *Efficient Word Lattice Generation for Joint Word Segmentation and POS Tagging in Japanese*  
Nobuhiro Kaji and Masaru Kitsuregawa
- 14:20–14:45 *A Simple Approach to Unknown Word Processing in Japanese Morphological Analysis*  
Ryohei Sasano, Sadao Kurohashi and Manabu Okumura
- 14:45–15:10 *Chinese Word Segmentation by Mining Maximized Substrings*  
Mo Shen, Daisuke Kawahara and Sadao Kurohashi

**October 15, 2013 (Tuesday) (continued)**

**POS Tagging and Parsing (Room 131 + 132)**

- 13:30–13:55 *Capturing Long-distance Dependencies in Sequence Models: A Case Study of Chinese Part-of-speech Tagging*  
Weiwei Sun, Xiaochang Peng and Xiaojun Wan
- 13:55–14:20 *Exploring Semantic Information in Hindi WordNet for Hindi Dependency Parsing*  
Sambhav Jain, Naman Jain, Aniruddha Tammewar, Riyaz Ahmad Bhat and Dipti Sharma
- 14:20–14:45 *Towards Robust Cross-Domain Domain Adaptation for Part-of-Speech Tagging*  
Tobias Schnabel and Hinrich Schütze
- 14:45–15:10 *Dependency Parsing for Identifying Hungarian Light Verb Constructions*  
Veronika Vincze, János Zsibrita and István Nagy T.

**Dialogue and Dialogue Systems (Room 133 + 134)**

- 13:30–13:55 *Written Dialog and Social Power: Manifestations of Different Types of Power in Dialog Behavior*  
Vinodkumar Prabhakaran and Owen Rambow
- 13:55–14:20 *Evaluation of the Scusi? Spoken Language Interpretation System – A Case Study*  
Thomas Kleinbauer, Ingrid Zukerman and Su Nam Kim
- 14:20–14:45 *A Noisy Channel Approach to Error Correction in Spoken Referring Expressions*  
Su Nam Kim, Ingrid Zukerman, Thomas Kleinbauer and Farshid Zavareh
- 14:45–15:10 *Natural Language Query Refinement for Problem Resolution from Crowd-Sourced Semi-Structured Data*  
Rashmi Gangadharaiah and Balakrishnan Narayanaswamy
- 15:10-15:40 Coffee Break
- 15:40-17:45 Regular Papers

**October 15, 2013 (Tuesday) (continued)**

**Machine Translation I (Reception Hall)**

- 15:40–16:05 *Ensemble Triangulation for Statistical Machine Translation*  
Majid Razmara and Anoop Sarkar
- 16:05–16:30 *Robust Transliteration Mining from Comparable Corpora with Bilingual Topic Models*  
John Richardson, Toshiaki Nakazawa and Sadao Kurohashi
- 16:30–16:55 *SuMT: A Framework of Summarization and MT*  
Houda Bouamor, Behrang Mohit and Kemal Oflazer
- 16:55–17:15 *Tuning SMT with a Large Number of Features via Online Feature Grouping*  
Lemao Liu, Tiejun Zhao, Taro Watanabe and Eiichiro Sumita
- 17:15–17:35 *Multimodal Comparable Corpora as Resources for Extracting Parallel Data: Parallel Phrases Extraction*  
Haithem Afli, Loïc Barrault and Holger Schwenk

**Information Extraction II (Room 141 + 142)**

- 15:40–16:05 *Bootstrapping Large-scale Named Entities using URL-Text Hybrid Patterns*  
Chao Zhang, Shiqi Zhao and Haifeng Wang
- 16:05–16:30 *Feature-Rich Segment-Based News Event Detection on Twitter*  
Yanxia Qin, Yue Zhang, Min Zhang and Dequan Zheng
- 16:30–16:55 *Building Chinese Event Type Paradigm Based on Trigger Clustering*  
Xiao Ding, Bing Qin and Ting Liu
- 16:55–17:20 *Chinese Named Entity Abbreviation Generation Using First-Order Logic*  
Huan Chen, Qi Zhang, Jin Qian and Xuanjing Huang
- 17:20–17:45 *Full-coverage Identification of English Light Verb Constructions*  
István Nagy T., Veronika Vincze and Richárd Farkas

**October 15, 2013 (Tuesday) (continued)**

**Recent NLP Applications I (Room 131 + 132)**

- 15:40–16:05 *Detecting Deceptive Opinions with Profile Compatibility*  
Vanessa Wei Feng and Graeme Hirst
- 16:05–16:30 *Behind the Times: Detecting Epoch Changes using Large Corpora*  
Octavian Popescu and Carlo Strapparava
- 16:30–16:55 *How Noisy Social Media Text, How Different Social Media Sources?*  
Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay and Li Wang
- 16:55–17:20 *Who Had the Upper Hand? Ranking Participants of Interactions Based on Their Relative Power*  
Vinodkumar Prabhakaran, Ajita John and Dorée D. Seligmann
- 17:20–17:45 *Readability Indices for Automatic Evaluation of Text Simplification Systems: A Feasibility Study for Spanish*  
Sanja Štajner and Horacio Saggion

**Language Resources I (Room 133 + 134)**

- 15:40–16:05 *Weasels, Hedges and Peacocks: Discourse-level Uncertainty in Wikipedia Articles*  
Veronika Vincze
- 16:05–16:30 *Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia*  
Fahd Alotaibi and Mark Lee
- 16:30–16:55 *Ranking Translation Candidates Acquired from Comparable Corpora*  
Rima Harastani, Béatrice Daille and Emmanuel Morin
- 16:55–17:20 *Using the Semantic-Syntactic Interface for Reliable Arabic Modality Annotation*  
Rania Al-Sabbagh, Jana Diesner and Roxana Girju
- 17:20–17:45 *Mapping Rules for Building a Tunisian Dialect Lexicon and Generating Corpora*  
Rahma Boujelbane, Mariem Ellouze Khemekhem and Lamia Hadrich Belguith
- 19:30-21:30 Reception (ANA Crowne Plaza Hotel Grand Court Nagoya)

**October 16, 2013 (Wednesday)**

09:00-10:15 Regular Papers

**Machine Translation II (Reception Hall East)**

09:00–09:25 *Hypothesis Refinement Using Agreement Constraints in Machine Translation*  
Ankur Gandhe and Rashmi Gangadharaiah

09:25–09:50 *Scalable Variational Inference for Extracting Hierarchical Phrase-based Translation Rules*  
Baskaran Sankaran, Gholamreza Haffari and Anoop Sarkar

09:50–10:15 *A Topic-Triggered Language Model for Statistical Machine Translation*  
Heng Yu, Jinsong Su, Yajuan Lv and Qun Liu

**Opinion Mining I (Reception Hall West)**

09:00–09:25 *Reserved Self-training: A Semi-supervised Sentiment Classification Method for Chinese Microblogs*  
Zhiguang Liu, Xishuang Dong, Yi Guan and Jinfeng Yang

09:25–09:50 *Enhancing Lexicon-Based Review Classification by Merging and Revising Sentiment Dictionaries*  
Heeryon Cho, Jong-Seok Lee and Songkuk Kim

09:50–10:15 *Exploring the Effects of Word Roots for Arabic Sentiment Analysis*  
Shereen Oraby, Yasser El-Sonbaty and Mohamad Abou El-Nasr

**October 16, 2013 (Wednesday) (continued)**

**Semantic Processing I (Room 131 + 132)**

- 09:00–09:25 *Topical Key Concept Extraction from Folksonomy*  
Han Xue, Bing Qin, Ting Liu and Chao Xiang
- 09:25–09:50 *Uncovering Distributional Differences between Synonyms and Antonyms in a Word Space Model*  
Silke Scheible, Sabine Schulte im Walde and Sylvia Springorum
- 09:50–10:15 *Multilingual Word Sense Disambiguation Using Wikipedia*  
Bharath Dandala, Rada Mihalcea and Razvan Bunescu

**Statistical and ML Language Modeling I (Room 133 + 134)**

- 09:00–09:25 *Semantic v.s. Positions: Utilizing Balanced Proximity in Language Model Smoothing for Information Retrieval*  
Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv and Xiaoming Li
- 09:25–09:50 *An Unsupervised Parameter Estimation Algorithm for a Generative Dependency N-gram Language Model*  
Chenchen Ding and Mikio Yamamoto
- 09:50–10:15 *Learning a Product of Experts with Elitist Lasso*  
Mengqiu Wang and Christopher D. Manning
- 10:15-10:45 Coffee Break
- 10:45-12:00 Regular Papers

**October 16, 2013 (Wednesday) (continued)**

**Information Extraction III / Question Answering (Reception Hall East)**

10:45–11:10 *Learning Efficient Information Extraction on Heterogeneous Texts*  
Henning Wachsmuth, Benno Stein and Gregor Engels

11:10–11:35 *TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction*  
Adrien Bougouin, Florian Boudin and Béatrice Daille

11:35–12:00 *Understanding the Semantic Intent of Natural Language Query*  
Juan Xu, Qi Zhang and Xuanjing Huang

**Opinion Mining II (Reception Hall West)**

10:45–11:10 *Sentiment Classification for Movie Reviews in Chinese Using Parsing-based Methods*  
Wen-Juan Hou and Chuang-Ping Chang

11:10–11:35 *Sentiment Aggregation using ConceptNet Ontology*  
Subhabrata Mukherjee and Sachindra Joshi

11:35–12:00 *Detecting Cyberbullying Entries on Informal School Websites Based on Category Relevance Maximization*  
Taisei Nitta, Fumito Masui, Michal Ptaszynski, Yasutomo Kimura, Rafal Rzepka and Kenji Araki

**Semantic Processing II (Room 131 + 132)**

10:45–11:10 *A Lexicon-based Investigation of Research Issues in Japanese Factuality Analysis*  
Kazuya Narita, Junta Mizuno and Kentaro Inui

11:10–11:35 *A Hierarchical Semantics-Aware Distributional Similarity Scheme*  
Shuqi Sun, Ke Sun, Shiqi Zhao, Haifeng Wang, Muyun Yang and Sheng Li

11:35–12:00 *Labeled Alignment for Recognizing Textual Entailment*  
Xiaolin Wang, Hai Zhao and Bao-Liang Lu



**October 16, 2013 (Wednesday) (continued)**

**Statistical and ML Language Modeling II (Room 133 + 134)**

10:45–11:10 *Context-Based Chinese Word Segmentation using SVM Machine-Learning Algorithm without Dictionary Support*  
Chia-ming Lee and Chien-Kang Huang

11:10–11:35 *A Common Case of Jekyll and Hyde: The Synergistic Effect of Using Divided Source Training Data for Feature Augmentation*  
Yan Song and Fei Xia

11:35–12:00 *Detecting Polysemy in Hard and Soft Cluster Analyses of German Preposition Vector Spaces*  
Sylvia Springorum, Sabine Schulte im Walde and Jason Utt

12:00–13:10 Lunch

13:10–15:30 Short Papers

**Recent NLP Applications / Text Summarization / Opinion Mining (Reception Hall East)**

13:10–13:30 *Generalized Abbreviation Prediction with Negative Full Forms and Its Application on Improving Chinese Web Search*  
Xu Sun, Wenjie Li, Fanqi Meng and Houfeng Wang

13:30–13:50 *Prosody-Based Unsupervised Speech Summarization with Two-Layer Mutually Reinforced Random Walk*  
Sujay Kumar Jauhar, Yun-Nung Chen and Florian Metze

13:50–14:10 *Mining the Gaps: Towards Polynomial Summarization*  
Marina Litvak and Natalia Vanetik

14:10–14:30 *Detecting Domain Dedicated Polar Words*  
Raksha Sharma and Pushpak Bhattacharyya

14:30–14:50 *Can I Hear You? Sentiment Analysis on Medical Forums*  
Tanveer Ali, David Schramm, Marina Sokolova and Diana Inkpen

14:50–15:10 *Construction of Emotional Lexicon Using Potts Model*  
Braja Gopal Patra, Hiroya Takamura, Dipankar Das, Manabu Okumura and Sivaji Bandyopadhyay

15:10–15:30 *Suicidal Tendencies: The Automatic Classification of Suicidal and Non-Suicidal Lyricists Using NLP*  
Matthew Mulholland and Joanne Quinn

**October 16, 2013 (Wednesday) (continued)**

**Language Resources / Information Extraction / Text Mining (Reception Hall West)**

- 13:10–13:30 *Unsupervised Word Class Induction for Under-resourced Languages: A Case Study on Indonesian*  
Meladel Mistica, Jey Han Lau and Timothy Baldwin
- 13:30–13:50 *An Efficient Active Learning Framework for New Relation Types*  
Lisheng Fu and Ralph Grishman
- 13:50–14:10 *Parsing Dependency Paths to Identify Event-Argument Relations*  
Seung-Cheol Baek and Jong Park
- 14:10–14:30 *Augmentable Paraphrase Extraction Framework*  
MeiHua Chen, YiChun Chen, ShihTing Huang and Jason S. Chang
- 14:30–14:50 *Automatic Prediction of Evidence-based Recommendations via Sentence-level Polarity Classification*  
Abeed Sarker, Diego Mollá-Aliod and Cécile Paris
- 14:50–15:10 *Clustering Microtext Streams for Event Identification*  
Jie Yin
- 15:10–15:30 *Automatic Corpora Construction for Text Classification*  
Dandan Wang, Qingcai Chen, Xiaolong Wang and Bingyang Yu

**Information Retrieval / NLP for Educational Applications / Phonology and Morphology (Room 131 + 132)**

- 13:10–13:30 *Learning to Generate Diversified Query Interpretations using Biconvex Optimization*  
Ramakrishna Bairi, Ambha A and Ganesh Ramakrishnan
- 13:30–13:50 *Learning Based Approaches for Vietnamese Question Classification Using Keywords Extraction from the Web*  
Dang Tran, Cuong Chu, Son Pham and Minh Nguyen
- 13:50–14:10 *Detecting Bot-Answerable Questions in Ubuntu Chat*  
David Uthus and David Aha
- 14:10–14:30 *Alignment-based Annotation of Proofreading Texts toward Professional Writing Assistance*  
Ngan Nguyen and Yusuke Miyao

**October 16, 2013 (Wednesday) (continued)**

14:30–14:50 *Toward Automatic Processing of English Metalanguage*  
Shomir Wilson

14:50–15:10 *On the Effectiveness of Using Syntactic and Shallow Semantic Tree Kernels for Automatic Assessment of Essays*  
Yllias Chali and Sadid A. Hasan

15:10–15:30 *Little by Little: Semi Supervised Stemming through Stem Set Minimization*  
Vasudevan N and Pushpak Bhattacharyya

**Semantic Processing / Pragmatics and Discourse (Room 133 + 134)**

13:10–13:30 *What Information is Helpful for Dependency Based Semantic Role Labeling*  
Yanyan Luo, Kevin Duh and Yuji Matsumoto

13:30–13:50 *Classifying Taxonomic Relations between Pairs of Wikipedia Articles*  
Or Biran and Kathleen McKeown

13:50–14:10 *A Rule System for Chinese Time Entity Recognition by Comprehensive Linguistic Study*  
Hongzhi Xu and Chu-Ren Huang

14:10–14:30 *Financial Sentiment Analysis for Risk Prediction*  
Chuan-Ju Wang, Ming-Feng Tsai, Tse Liu and Chin-Ting Chang

14:30–14:50 *Sense Disambiguation: From Natural Language Words to Mathematical Terms*  
Minh-Quoc Nghiem, Giovanni Yoko Kristianto, Goran Topic and Akiko Aizawa

14:50–15:10 *Adapting a State-of-the-art Anaphora Resolution System for Resource-poor Language*  
Utpal Sikdar, Asif Ekbal, Sriparna Saha, Olga Uryupina and Massimo Poesio

15:10–15:30 *Chinese Event Coreference Resolution: Understanding the State of the Art*  
Chen Chen and Vincent Ng

15:30-17:00 Poster Presentations and System Demonstrations

*A Two-Step Named Entity Recognizer for Open-Domain Search Queries*  
Andreas Eiselt and Alejandro Figueroa

October 16, 2013 (Wednesday) (continued)

*A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction*

Florian Boudin

*Translating Chinese Unknown Words by Automatically Acquired Templates*

Ming-Hong Bai, Yu-Ming Hsieh, Keh-Jiann Chen and Jason S. Chang

*Multilingual Lexicon Bootstrapping - Improving a Lexicon Induction System Using a Parallel Corpus*

Patrick Ziering, Lonneke van der Plas and Hinrich Schütze

*Mining Japanese Compound Words and Their Pronunciations from Web Pages and Tweets*

Xianchao Wu

*A Factoid Question Answering System Using Answer Pattern Matching*

Nagehan Pala Er and Ilyas Cicekli

*Chinese Short Text Classification Based on Domain Knowledge*

Xiao Feng, Yang Shen, Chengyong Liu, Wei Liang and Shuwu Zhang

*Applying Graph-based Keyword Extraction to Document Retrieval*

Youngsam Kim, Munhyong Kim, Andrew Cattle, Julia Otmakhova, Suzi Park and Hyopil Shin

*Semi-supervised Classification of Twitter Messages for Organization Name Disambiguation*

Shu Zhang, Jianwei Wu, Dequan Zheng, Yao Meng and Hao Yu

*Word in a Dictionary is used by Numerous Users*

Eiji Aramaki, Sachiko Maskawa, Mai Miyabe, Mizuki Morita and Sachi Yasuda

*Extracting Evaluative Conditions from Online Reviews: Toward Enhancing Opinion Mining*

Yuki Nakayama and Atsushi Fujii

*Cognate Production using Character-based Machine Translation*

Lisa Beinborn, Torsten Zesch and Iryna Gurevych

*An Empirical Study of Combining Multiple Models in Bengali Question Classification*

Somnath Banerjee and Sivaji Bandyopadhyay

October 16, 2013 (Wednesday) (continued)

*A Two-Stage Classifier for Sentiment Analysis*

Dai Quoc Nguyen, Dat Quoc Nguyen and Son Bao Pham

*Exploiting User Search Sessions for the Semantic Categorization of Question-like Informational Search Queries*

Alejandro Figueroa and Guenter Neumann

*Influence of Part-of-Speech and Phrasal Category Universal Tag-set in Tree-to-Tree Translation Models*

Francisco Oliveira, Derek F. Wong, Lidia S. Chao, Liang Tian and Liangye He

*Interest Analysis using PageRank and Social Interaction Content*

Chung-chi Huang and Lun-Wei Ku

*Time Series Topic Modeling and Bursty Topic Detection of Correlated News and Twitter*

Daichi Koike, Yusuke Takahashi, Takehito Utsuro, Masaharu Yoshioka and Noriko Kando

*A Distant Supervision Approach for Identifying Perspectives in Unstructured User-Generated Text*

Attapol Thamrongrattanarit, Colin Pollock, Benjamin Goldenberg and Jason Fennell

*An Approach of Hybrid Hierarchical Structure for Word Similarity Computing by HowNet*

Jiangming Liu, Jinan Xu and Yujie Zhang

*Extracting Causes of Emotions from Text*

Alena Neviarouskaya and Masaki Aono

*Automated Grammar Correction Using Hierarchical Phrase-Based Statistical Machine Translation*

Bibek Behera and Pushpak Bhattacharyya

*Finding Dependency Parsing Limits over a Large Spanish Corpus*

Muntsa Padró, Miguel Ballesteros, Héctor Martínez and Bernd Bohnet

*High Quality Dependency Selection from Automatic Parses*

Gongye Jin, Daisuke Kawahara and Sadao Kurohashi

*Building Specialized Bilingual Lexicons Using Word Sense Disambiguation*

Dhouha Bouamor, Nasredine Semmar and Pierre Zweigenbaum

**October 16, 2013 (Wednesday) (continued)**

*Predicate Argument Structure Analysis using Partially Annotated Corpora*

Koichiro Yoshino, Shinsuke Mori and Tatsuya Kawahara

*Statistical Dialogue Management using Intention Dependency Graph*

Koichiro Yoshino, Shinji Watanabe, Jonathan Le Roux and John R. Hershey

*Repairing Incorrect Translation with Examples*

Junguo Zhu, Muyun Yang, Sheng Li and Tiejun Zhao

*Phrase-based Parallel Fragments Extraction from Comparable Corpora*

Xiaoyin Fu, Wei Wei, Shixiang Lu, Zhenbiao Chen and Bo Xu

*A Hybrid Approach for Anaphora Resolution in Hindi*

Praveen Dakwale, Vandan Mujadia and Dipti M Sharma

*Structure Cognizant Pseudo Relevance Feedback*

Arjun Atreya V, Yogesh Kakde, Pushpak Bhattacharyya and Ganesh Ramakrishnan

*Cross-Domain Answer Ranking using Importance Sampling*

Anders Johannsen and Anders Søgaard

*Morphological Analysis of Tunisian Dialect*

Inès Zribi, Mariem Ellouze Khemakhem and Lamia Hadrich Belguith

*Disambiguating Explicit Discourse Connectives without Oracles*

Anders Johannsen and Anders Søgaard

*Updating Rare Term Vector Replacement*

Tobias Berka and Marian Vajteršić

*Statistical Morphological Analyzer for Hindi*

Deepak Kumar Malladi and Prashanth Mannem

*Induction of Root and Pattern Lexicon for Unsupervised Morphological Analysis of Arabic*

Bilal Khaliq and John Carrol

**October 16, 2013 (Wednesday) (continued)**

*Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text*

Minh Quang Nhat Pham, Minh Le Nguyen and Akira Shimazu

*Using Transliteration of Proper Names from Arabic to Latin Script to Improve English-Arabic Word Alignment*

Nasredine Semmar and Houda Saadane

*A Semi-Supervised Method for Arabic Word Sense Disambiguation Using a Weighted Directed Graph*

Laroussi Merhbene, Anis Zouaghi and Mounir Zrigui

*Incremental Segmentation and Decoding Strategies for Simultaneous Translation*

Mahsa Yarmohammadi, Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore and Baskaran Sankaran

*Two Case Studies on Translating Pronouns in a Deep Syntax Framework*

Michal Novák, Zdenek Zabokrtsky and Anna Nedoluzhko

*Bootstrapping Phrase-based Statistical Machine Translation via WSD Integration*

Hien Vu Huy, Phuong-Thai Nguyen, Tung-Lam Nguyen and M.L Nguyen

*Orthographic and Morphological Processing for Persian-to-English Statistical Machine Translation*

Mohammad Sadegh Rasooli, Ahmed El Kholy and Nizar Habash

*Interoperability between Service Composition and Processing Pipeline: Case Study on the Language Grid and UIMA*

Trang Mai Xuan, Yohei Murakami, Donghui Lin and Toru Ishida

*Improving Calculation of Contextual Similarity for Constructing a Bilingual Dictionary via a Third Language*

Takashi Tsunakawa, Yosuke Yamamoto and Hiroyuki Kaji

*Two-Stage Pre-ordering for Japanese-to-English Statistical Machine Translation*

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh and Masaaki Nagata

*Grammatical Error Correction Using Feature Selection and Confidence Tuning*

Yang Xiang, Yaoyun Zhang, Xiaolong Wang, Chongqiang Wei, Wen Zheng, Xiaoqiang Zhou, Yuxiu Hu and Yang Qin

*An Online Algorithm for Learning over Constrained Latent Representations using Multiple Views*

Ann Clifton, Max Whitney and Anoop Sarkar

**October 16, 2013 (Wednesday) (continued)**

*Synonym Acquisition Using Bilingual Comparable Corpora*

Daniel Andrade, Masaaki Tsuchida, Takashi Onishi and Kai Ishikawa

*Exploring Verb Frames for Sentence Simplification in Hindi*

Ankush Soni, Sambhav Jain and Dipti Misra Sharma

*Dirichlet Processes for Joint Learning of Morphology and PoS Tags*

Burcu Can and Suresh Manandhar

*Parser Accuracy in Quality Estimation of Machine Translation: A Tree Kernel Approach*

Rasoul Samad Zadeh Kaljahi, Jennifer Foster, Raphael Rubino, Johann Roturier and Fred Hollowood

*Attribute Relation Extraction from Template-inconsistent Semi-structured Text by Leveraging Site-level Knowledge*

Yang Liu, Fang Liu, Siwei Lai, Kang Liu, Guangyou Zhou and Jun Zhao

*Optimum Parameter Selection for K.L.D. Based Authorship Attribution in Gujarati*

Parth Mehta and Prasenjit Majumder

*Modeling User Leniency and Product Popularity for Sentiment Classification*

Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji and Masaru Kitsuregawa

*A Generalized LCS Algorithm and Its Application to Corpus Alignment*

Jin-Dong Kim

*Semantic Naïve Bayes Classifier for Document Classification*

How Jing, Yu Tsao, Kuan-Yu Chen and Hsin-Min Wang

*Cluster-based Web Summarization*

Yves Petinot, Kathleen McKeown and Kapil Thadani

*Automated Activity Recognition in Clinical Documents*

Camilo Thorne, Marco Montali, Diego Calvanese, Elena Cardillo and Claudio Eccher

*Large-Scale Text Collection for Unwritten Languages*

Florian R. Hanke and Steven Bird



**October 16, 2013 (Wednesday) (continued)**

*A Self-learning Template Approach for Recognizing Named Entities from Web Text*  
Qian Liu, Bingyang Liu, Dayong Wu, Yue Liu and Xueqi Cheng

18:00-21:00 Banquet (Port of Nagoya Public Aquarium)

**October 17, 2013 (Thursday)**

09:00-10:00 Keynote Speech - Roberto Navigli (Sapienza University of Rome) (Reception Hall)

10:00-10:30 Coffee Break

10:30-12:30 Short Papers

**Machine Translation (Reception Hall)**

10:30–10:50 *Accurate Parallel Fragment Extraction from Quasi-Comparable Corpora using Alignment Model and Translation Lexicon*  
Chenhui Chu, Toshiaki Nakazawa and Sadao Kurohashi

10:50–11:10 *Meta-level Statistical Machine Translation*  
Sajad Ebrahimi, Kourosh Meshgi, Shahram Khadivi and Mohammad Ebrahim Shiri Ahmad Abady

11:10–11:30 *Bayesian Induction of Bracketing Inversion Transduction Grammars*  
Markus Saers and Dekai Wu

11:30–11:50 *Estimating the Quality of Translated User-Generated Content*  
Raphael Rubino, Jennifer Foster, Rasoul Samad Zadeh Kaljahi, Johann Roturier and Fred Hollowood

11:50–12:10 *Selective Combination of Pivot and Direct Statistical Machine Translation Models*  
Ahmed El Kholy, Nizar Habash, Gregor Leusch, Evgeny Matusov and Hassan Sawaf

12:10–12:30 *Multiword Expressions in the Context of Statistical Machine Translation*  
Mahmoud Ghoneim and Mona Diab

October 17, 2013 (Thursday) (continued)

**Recent NLP Applications / Dialogue and Dialogue Systems (Room 141 + 142)**

- 10:30–10:50 *Uncertainty Detection for Natural Language Watermarking*  
György Szarvas and Iryna Gurevych
- 10:50–11:10 *KySS 1.0: a Framework for Automatic Evaluation of Chinese Input Method Engines*  
Zhongye Jia and Hai Zhao
- 11:10–11:30 *Automatic Extraction of Social Networks from Literary Text: A Case Study on Alice in Wonderland*  
Apoorv Agarwal, Anup Kotalwar and Owen Rambow
- 11:30–11:50 *Using the Web to Train a Mobile Device Oriented Japanese Input Method Editor*  
Xianchao Wu, Rixin Xiao and Xiaoxin Chen
- 11:50–12:10 *A Novel Approach Towards Incorporating Context Processing Capabilities in NLIDB System*  
Arjun Akula, Rajeev Sangal and Radhika Mamidi
- 12:10–12:30 *Iterative Development and Evaluation of a Social Conversational Agent*  
Annika Silvervarg and Arne Jönsson

**POS Tagging and Parsing / Syntax and Semantics / Information Extraction (Room 131 + 132)**

- 10:30–10:50 *A Hybrid Morphological Disambiguation System for Turkish*  
Mucahid Kutlu and Ilyas Cicekli
- 10:50–11:10 *A Dynamic Confusion Score for Dependency Arc Labels*  
Sambhav Jain and Bhasha Agrawal
- 11:10–11:30 *Increasing the Quality and Quantity of Source Language Data for Unsupervised Cross-Lingual POS Tagging*  
Long Duong, Paul Cook, Steven Bird and Pavel Pecina
- 11:30–11:50 *Towards the Annotation of Penn TreeBank with Information Structure*  
Bernd Bohnet, Alicia Burga and Leo Wanner
- 11:50–12:10 *Constituency and Dependency Relationship from a Tree Adjoining Grammar and Abstract Categorical Grammars Perspective*  
Aleksandre Maskharashvili and Sylvain Pogodalla

**October 17, 2013 (Thursday) (continued)**

12:10–12:30 *Named Entity Extraction using Information Distance*  
Sangameshwar Patil, Sachin Pawar and Girish Palshikar

**Statistical and ML Language Modeling (Room 133 + 134)**

10:30–10:50 *Feature-based Neural Language Model and Chinese Word Segmentation*  
Mairgup Mansur, Wenzhe Pei and Baobao Chang

10:50–11:10 *Human-Computer Interactive Chinese Word Segmentation: An Adaptive Dirichlet Process Mixture Model Approach*  
Tongfei Chen, Xiaojun Zou, Weimeng Zhu and Junfeng Hu

11:10–11:30 *Effect of Non-linear Deep Architecture in Sequence Labeling*  
Mengqiu Wang and Christopher D. Manning

11:30–11:50 *Case Study of Model Adaptation: Transfer Learning and Online Learning*  
Kenji Imamura

11:50–12:10 *Source and Translation Classification using Most Frequent Words*  
Zahurul Islam and Armin Hoenen

12:10–12:30 *Comparison of Algorithmic and Human Assessments of Sentence Similarity*  
John Mersch and R. Raymond Lang

12:30-14:00 Lunch

14:00-16:05 Regular Papers

**October 17, 2013 (Thursday) (continued)**

**Information Extraction IV (Reception Hall)**

- 14:05–14:30 *Effective Selectional Restrictions for Unsupervised Relation Extraction*  
Alan Akbik, Larysa Visengeriyeva, Johannes Kirschnick and Alexander Löser
- 14:30–14:55 *Bootstrapping Semantic Lexicons for Technical Domains*  
Patrick Ziering, Lonneke van der Plas and Hinrich Schütze
- 14:55–15:20 *Long-Distance Time-Event Relation Extraction*  
Alessandro Moschitti, Siddharth Patwardhan and Chris Welty
- 15:20–15:45 *Unsupervised Extraction of Attributes and Their Values from Product Description*  
Keiji Shinzato and Satoshi Sekine
- 15:45–16:05 *Stance Classification of Ideological Debates: Data, Models, Features, and Constraints*  
Kazi Saidul Hasan and Vincent Ng

**Language Resources II / Recent NLP Applications II (Room 141 + 142)**

- 14:05–14:30 *University Entrance Examinations as a Benchmark Resource for NLP-based Problem Solving*  
Yusuke Miyao and Ai Kawazoe
- 14:30–14:55 *Linguistically Aware Coreference Evaluation Metrics*  
Chen Chen and Vincent Ng
- 14:55–15:20 *An Empirical Assessment of Contemporary Online Media in Ad-Hoc Corpus Creation for Social Events*  
Kanika Narang, Seema Nagar, Sameep Mehta, L V Subramaniam and Kuntal Dey
- 15:20–15:45 *Diagnosing Causes of Reading Difficulty using Bayesian Networks*  
Pascual Martínez-Gómez and Akiko Aizawa
- 15:45–16:05 *Word Co-occurrence Counts Prediction for Bilingual Terminology Extraction from Comparable Corpora*  
Amir Hazem and Emmanuel Morin

**October 17, 2013 (Thursday) (continued)**

**Text Summarization / NLP for Educational Applications (Room 131 + 132)**

- 14:05–14:30 *Measuring the Effect of Discourse Relations on Blog Summarization*  
Shamima Mithun and Leila Kosseim
- 14:30–14:55 *Supervised Sentence Fusion with Single-Stage Inference*  
Kapil Thadani and Kathleen McKeown
- 14:55–15:20 *Detecting and Correcting Learner Korean Particle Omission Errors*  
Ross Israel, Markus Dickinson and Sun-Hee Lee
- 15:20–15:45 *Automatic Identification of Learners' Language Background Based on Their Writing in Czech*  
Katsiaryna Aharodnik, Marco Chang, Anna Feldman and Jirka Hana
- 16:05-16:35 Coffee Break
- 16:35-17:15 Best Papers (Reception Hall)
- 17:15-17:25 Future Conferences (Reception Hall)
- 17:25-17:35 Closing (Reception Hall)

# Semi-Supervised Answer Extraction from Discussion Forums

Rose Catherine, Rashmi Gangadharaiah, Karthik Visweswariah, Dinesh Raghu

IBM Research  
Bangalore, India

{rosecatherinek, rashgang, v-karthik, diraghu1} @in.ibm.com

## Abstract

Mining online discussions to extract answers is an important research problem. Methods proposed in the past used supervised classifiers trained on labeled data. But, collecting training data for each target forum is labor intensive and time consuming, thus limiting their deployment. A recent approach had proposed to extract answers in an unsupervised manner, by taking cues from their repetitions. This assumption however, does not hold true in many cases. In this paper, we propose two semi-supervised methods for extracting answers from discussions, which utilize the large amount of unlabeled data available, alongside a very small training set to obtain improved accuracies. We show that it is possible to boost the performance by introducing a related, but parallel task of identifying acknowledgments to the answers. The accuracy achieved by our approaches surpass the baselines by a wide margin, as shown by our experiments.

## 1 Introduction

Online discussion forums, also known as community question answering (CQA) sites, are internet sites that provide a medium for users to discuss and share information on a wide range of topics. Due to their vast popularity, gradually, they have aggregated a massive collection of discussion data. Mining such forums have numerous applications such as improving question-answer (QA) retrieval (Cong et al., 2008), learning important insights like features of products that are drawing negative reviews (Lakkaraju et al., 2011) or discovering longstanding unresolved severe technical issues (Gangadharaiah and Catherine, 2012) etc. For this reason, substantial research effort has been directed at mining discussions, in recent

times. In this paper, we focus on the specific problem of extracting answers from these discussions.

In forums, typically a user starts a discussion by posting a question to which multiple members of the forum suggest answers. The discussion evolves into a complex multi-party conversation as the question gets refined, with additional details specified, clarifications sought, multiple answers provided, frequent digressions, and occasional follow-up discussions and acknowledgments, altogether spanning several pages. Answers easily get buried deep within this and locating them automatically is far from straightforward.

In this paper, we propose two semi-supervised approaches that require only a very small amount of training data (only 3 manually tagged discussion threads) and achieve high accuracy levels by using the available unlabeled data. With this, we eliminate the need to collect vast amounts of training data, thus aiding faster deployment for new domains. Specifically, our contributions are:

- *A semi-supervised answer extraction method for discussions:* This paper makes the first attempt at extracting answers from discussions in a semi-supervised manner. We show how existing features can be engineered into a co-training framework to accomplish this.
- *A parallel co-training method to leverage acknowledgments for improved answer extraction accuracy:* We motivate and demonstrate that it is possible to improve the performance tremendously by introducing a related task of identifying acknowledgments in the discussions, which we run as a parallel task alongside the main answer extraction task (Section 5).
- We demonstrate that with a very small training data and by using the available unlabeled data, it is possible to extract answers from forums with an accuracy that is substantially better than extracting them in an unsupervised manner or in a fully supervised setting.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 sets the terminology and introduces the co-training framework, which is used throughout this paper. Section 4 details how the co-training framework can be applied to the answer extraction task. Section 5 introduces the acknowledgment extraction task in a parallel co-training framework. Experiments and results are discussed in Section 6 followed by conclusions in Section 7.

## 2 Related work

Research in the area of extracting question and answers from online forums, has grown considerably. Almost all approaches proposed so far for this task are supervised learning methods. Ding et al. (2008), Kim et al. (2010), Raghavan et al. (2010) and Kim et al. (2012) employed Conditional Random Fields, Hong and Davison (2009), Huang et al. (2007) and Catherine et al. (2012) used Support Vector Machines (SVM), Shrestha and McKeown (2004) learnt rules using Ripper, and Yang et al. (2009) used Struct SVMs for extracting answers. The obvious downside to these methods is that for any new domain or forum, substantial amounts of manually labeled training examples have to be collected. This is usually time consuming and costly. Gandhe et al. (2012) proposed an approach for adapting an answer extractor trained on one domain to another, by separating out the lexical characteristics of an answer from its domain relevance. However, learning the lexical characteristics still required a training set.

A recent work by Cong et al. (2008) proposed an unsupervised method using PageRank-style random walks on a graph representation of the discussion, with the hypothesis that inter-candidate similarities can improve accuracy of the answer extraction task. The intuition is that posts that bear more resemblance to other posts in the thread have higher chances of being answers. However, in a lot of discussion forums, especially those related to troubleshooting and problem resolution, we found that this assumption usually does not hold. An answer that was suggested earlier in the discussion is not usually suggested again – only new ones or a modification of the same would appear. A general observation here was that posts that had similar content as other posts were found to be others complaining about the same issue. This was also noted by Gandhe et al. (2012) and Catherine et al. (2012). Nevertheless, (Cong et al., 2008) is the

only work so far, that sought to extract answers without supervision.

One of the methods proposed in this paper that uses a parallel acknowledgment classification task, belongs to the family of Multi-Task Learning (MTL) (Caruana, 1997) since what is learned for each task is used to improve the other task. However, to the best of our knowledge, this is the first work that proposes a MTL-type answer classifier for forums in a semi-supervised setting. Cross-Training (Sarawagi et al., 2003) is a related methodology which improves classification performance on one taxonomy by accessing labels from another taxonomy for the *same* document. Our method differs because, the answer and acknowledgment labels are on *different* posts.

Some other closely related works are listed below; however, their focus is different from the task proposed in this paper. Jijkoun and de Rijke (2005) proposed a method to automatically extract question–answer pairs from FAQ pages using formatting cues. Since it is known that the entry following the question is definitely an answer, they did not have to classify the entries. Sarencheh et al. (2010) proposed a semi-automatic wrapper induction method for extracting different structural components of a discussion, like the time of posting, author name, content of the post etc. Answer retrieval is another closely related task where the emphasis is on retrieving the most relevant post (Xue et al., 2008). The scope of our paper, however, is limited to tagging posts in forum discussions as answers or not.

## 3 Preliminaries

### 3.1 Terminology and Scope

A discussion in an online forum is created when a user posts a question. Other members of the forum reply to this post or to other replies, thereby evolving the discussion. A sample discussion with 7 posts including 2 answers and 2 acknowledgments, is shown in Figure 1. In this paper, we use the terms *discussion* and *thread* (as in, a thread of discussion) interchangeably.

An answer is typically spread over multiple sentences within the same post, especially in the case of non-factoid answers. It would have been ideal, if the system extracted answers at the granularity of a sentence. However, the inter-annotator agreement<sup>1</sup> for answer sentences in our dataset (Section 6) was a mere 0.19. Hence, we extract answers at

<sup>1</sup>[http://en.wikipedia.org/wiki/Cohen's\\_kappa](http://en.wikipedia.org/wiki/Cohen's_kappa)

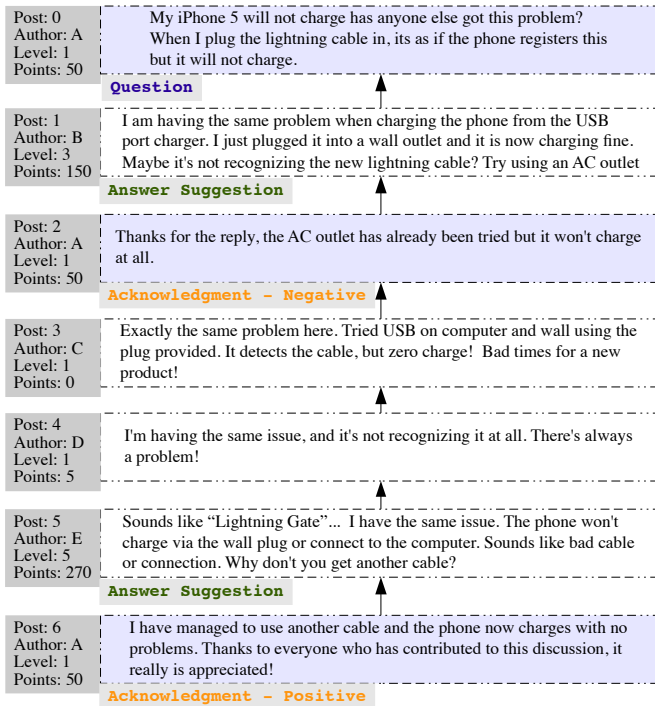


Figure 1: A Sample Discussion

a post level – a post is classified as an answer if any sentence within it suggests a solution.

Digressions are very common in such community question answering systems. We do not attempt to find these questions or separate out the sub-discussions. Question detection as well as disentangling multi-party discussions, is a well researched area (Cong et al., 2008; Elsner and Charniak, 2010), and is outside the scope of this paper. For the purposes of this paper, the first post is the question and we attempt to find answers to only this question. Answers to other questions within the discussion are negative examples.

### 3.2 Co-Training Methodology

Co-Training introduced by Blum and Mitchell (1998), is a general framework for semi-supervised classification, where the features for classifying each data point can be partitioned into two distinct sets or views. The views are such that either of them is sufficient to classify any data-point, had there been enough training examples.

The algorithm proceeds in two half-steps: in iteration  $i$ , the current set of labeled examples  $L^i$  (initially, a very small set) is used to train a classifier  $C_1$  that uses only one view  $v_1$  of each training instance and another classifier  $C_2$  that uses only view  $v_2$ .  $C_1$  and  $C_2$  are then used to classify the unlabeled points, and the most confident  $m$  predictions are moved from the unlabeled pool  $U$  to the set of labeled examples, which are used

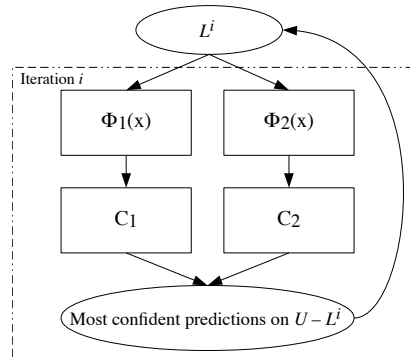


Figure 2: Co-Training Framework

for training in the  $i + 1^{th}$  iteration. Essentially, each classifier teaches the other by providing examples which the other would have misclassified. Figure 2 shows this workflow, where  $\Phi_1$  and  $\Phi_2$  are the feature vectors of the input corresponding to the two views  $v_1$  and  $v_2$ . The paper showed that when the two views are independent given the label of the data point (conditional independence), any initial weak predictor can be boosted to a high accuracy using unlabeled examples by co-training. This was empirically evaluated for a webpage classification task where  $v_1$  was the set of words in the webpage and  $v_2$ , the anchor texts of all links pointing to that page. Co-training framework is widely used in many text mining tasks like parsing (Sarkar, 2001), machine translation (Callison-Burch, 2002) and for creating parallel corpora (Callison-Burch and Osborne, 2003).

### 4 Answer Extraction by Co-Training: ANS CT Model

To apply the co-training framework to the task of answer extraction, we need two independent views of the data. Prior work in supervised answer extraction from forum discussions have reported good accuracies when using features constructed from the structure of the thread (Ding et al., 2008; Hong and Davison, 2009; Kim et al., 2010; Catherine et al., 2012). This provides us with one of the views, which we refer to as the STRUCT view.

Cong et al. (2008) had previously used pattern mining for the related task of question sentence extraction. Similarly, Jindal and Liu (2006) had used pattern mining for identifying comparative sentences in a supervised learning setting. We mine patterns on the sentences of the posts and employ it as the second view, which we refer to as the PATTERN view. The exact set of features are explained in the subsections below.



Note that we have used only a modest set of features in both STRUCT and PATTERN views, to highlight the effect of co-training in improving the answer extraction accuracy.

#### 4.1 STRUCT view

Compared to a general text document, discussion threads have a structure, which can be used to construct features for classification. The features that we use, referred to as STRUCT features henceforth, are listed in Table 1. All the features are eventually converted to binary attributes for the experiments, where numerical attributes are grouped into a suitable number of buckets. Each binary value corresponds to a dimension in the STRUCT view,  $\Phi_{struct}^i$ , which is 1 if that attribute-value was present in the post; else, is set to 0.

STRUCT Feature	Description
Author Rating	A forum specific value measuring the expertise of the author. Could be numerical (e.g. 50 points) or categorical (e.g. Expert).
Relative Post Position	The position of the post with respect to the thread. It is grouped into Beginning, Middle and End.
Post Rating	A measure of how informative the post is. Could be numerical (e.g. 50 votes) or categorical (e.g. Helpful).

Table 1: STRUCT features for a post

#### 4.2 PATTERN view

Consider the below snippets from different discussion threads. Some words have been intentionally masked to illustrate that it is possible to identify to a considerable extent, that these are answer suggestions from the structure of the sentence and without regard to the context or the question.

... You can see if X will solve it ...  
 ... Try resetting your X with the Y turned off and then turn it back on after the X is fully booted back up ...  
 ... Go to A -> B -> C and toggle the D mode ...  
 ... X is no longer supported by Y ...

The PATTERN view uses a pattern mining module, which mines the answer posts in  $L^i$  to discover the most frequent sequential patterns,  $FP^i$  for iteration  $i$ . Each such discovered pattern  $p \in FP^i$  corresponds to a dimension in the PATTERN view,  $\Phi_{pattern}^i$ , which is 1 if  $p$  matches (is sub-sequence of) any sentence of the post.

We implemented the PrefixSpan (Pei et al., 2001) algorithm for mining sequential patterns, but with the following modifications to contain the blow up in the number of patterns:

- Variable Minimum Support: the number of items in which a pattern appears is called its *support*, and minimum support, `min_sup` is an input parameter that determines whether a pattern is frequent enough. We set `min_sup` to  $\max(\text{min\_sup}_0, \text{frac} \times \text{numItems}^i)$ , where `frac` is a preset fraction, set to 0.03 in our experiments,  $\text{numItems}^i$  is the number of items being mined in iteration  $i$ , and `min_sup0` is a default minimum, set to 3 in our experiments.
- Pattern Length: only patterns of length at least `min_len`, set to 3 in our experiments, are acceptable.
- Item Gap: the items of a frequent pattern are sequential, but not consecutive, thus allowing PrefixSpan to pick items that are arbitrary number of items apart (`gap`). We constrain the gap between items of a pattern to a maximum of `max_gap`, set to again 3 in our experiments.

Posts are mined at a sentence level, for which we use OpenNLP<sup>2</sup> sentence detector.

##### 4.2.1 Text Pre-Processing

We found that using the exact words limited the number of frequent patterns that could be found. To minimize this problem, we used Part-Of-Speech (POS) tags of the words to:

- Replace all nouns with their POS tags.
- Replace all verbs with its root/stemmed (using Porter stemmer (Porter, 1980)) form and its POS tag. For example, `restarting` becomes `restart VBG`. We let PrefixSpan pick the verb-stem and/or the POS tag according to their support.

All words are lowercased. A discussion on the set of patterns that were detected is in Section 6.3.

## 5 Leveraging Acknowledgment Signals: ANS-ACK PCT Model

In this section, we motivate and introduce a related task of extracting acknowledgments in forum discussions and inducing signals from them to improve the accuracy of the answer extraction task.

Merriam-Webster<sup>3</sup> defines an acknowledgment as a recognition or favorable notice of an act or achievement. Acknowledgment is an inevitable component of any conversation, especially, when it evolves around seeking assistance. And so they find their place in forum discussions too. Consider the below snippets taken from replies by question

<sup>2</sup><http://opennlp.apache.org>

<sup>3</sup><http://www.merriam-webster.com>

authors. They are grouped according to their polarity – Positive, Negative and Neutral.

**Positive:** author reports that the suggestion solved the issue.

... Great! That solved it! Thanks a bunch ...  
... Thanks for your help. Finally got it working ...  
... Switching on X did the trick. Now I can Y without any problem ...  
... Thanks a lot guys. X solved my woes. I must have Y-ed it by mistake at some point ...

**Negative:** the suggestion did not solve the issue.

... That didn't help. Any other suggestions? ...  
... I tried that. It is still showing X ...  
... Getting the same X. Thanks anyway ...  
... Thanks for your advice. Unfortunately, it didn't help!

**Neutral:** it is not clear if the issue was solved, but the statement is an acknowledgment nevertheless

... Thanks for the reply ...  
... I will try that ...  
... Thanks for the helpful advice. Hope resetting X properly will fix my problem ...  
... I'm reinstalling X. Will keep you posted ...

Similar to the case of answer sentences in Section 4, the above examples can be easily identified as acknowledgments and it is fairly clear that the posts to which the above sentences are replies, are answer suggestions. Note that this can be determined without knowing the contents of the latter, if we can assume that the reply-to relation of the posts is known. This however is not always the case. In a small study conducted, we found that only 75% of forums displayed or had the required information in the html of the webpage for constructing the reply-to relation of the posts, out of 12 technical forums that we inspected. In the absence of this information, (Wang et al., 2011; Seo et al., 2009; Wang and Rosé, 2010) propose techniques to automatically recover the structure. For the purposes of this paper, we assume that the reply-to structure of the discussion thread is given.

ANS-ACK PCT (ANSWER ACKnowledgment Parallel Co-Training) aims to leverage signals from acknowledgment posts, to better identify answer posts. We cast this as another instance of semi-supervised learning task (another co-training instance) which runs in parallel to the main answer extraction instance of co-training. Hence the name, PARALLEL co-training.

It is worth listing down some of the design decisions for this choice of approach:

(i) There is no public dataset available to train an acknowledgment classifier. So, it is important to note here that the task of detecting acknowledgments cannot be fully supervised where a large

amount of training data is collected for the specific domain; this will defeat the entire purpose of semi-supervised answer extraction.

(ii) For the initial small training set required for the semi-supervised approach, we do not label additional threads. Instead, we create a training set from the initial training set of the answer extraction task by marking replies from the question author to posts that are answers, as positive examples. Other replies from the question author become negative examples. To avoid getting influenced by digressions, we do not consider replies from other authors.

(iii) The reader might suggest using acknowledgment as one of the views within the co-training instance of answer detection, instead of two parallel co-training instances. i.e. to mark all posts that have an acknowledgment as an answer in that view. Here, we would like to point out that acknowledgment is a strong indicator only when it is available. In other words, even if we learn to classify acknowledgment posts perfectly, it cannot classify all answer posts perfectly since not all answers are acknowledged. In our test set (Section 6), there were 559 answers, but only 173 of them had any reply from the question author (30.9%), of which only 72% were actually acknowledgments, as found through manual inspection (the rest had to do with refining the question, requesting clarification on the answer, etc.). So, the hope is to learn how to use the signal when it is available, and not rely on it exclusively by using it as one of the two views of answer co-training.

The acknowledgment extraction uses the same two views – STRUCT and PATTERN – for its co-training instance, similar to the ANS CT model of Section 4, to generate the views,  $^{ack}\Psi_{struct}^i$  and  $^{ack}\Psi_{pattern}^i$ , respectively. Except that here, positive examples are the posts that are acknowledgments, as obtained by Point (ii) above.

## 5.1 Parallel Co-Training for Answer Extraction

Parallel Co-Training is a method for semi-supervised learning where there are two (or more) co-training instances corresponding to different, but related learning tasks running side by side, where in iteration  $i$ , each task can induce features based on the current state of the system. i.e. using the outcome of iteration  $i - 1$  of other tasks. Figure 3 depicts Parallel Co-Training for the specific case of answer extraction, where:

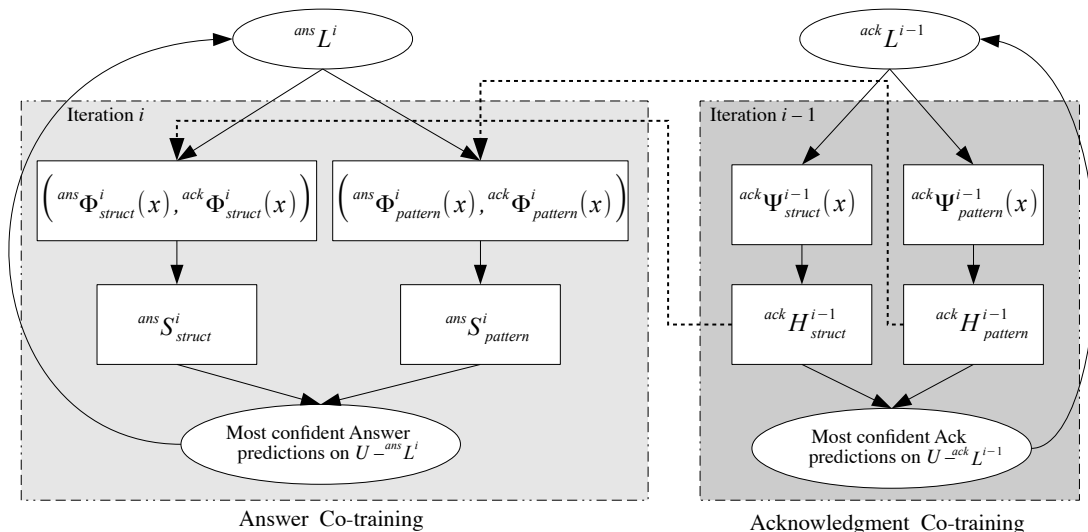


Figure 3: Parallel Co-Training Methodology

- Two tasks run in parallel:
  1. Answer Co-training: the main task which learns to classify each post as  $\text{ans}$  or  $\overline{\text{ans}}$ .
  2. Acknowledgment Co-training: the auxiliary task which learns to classify each post as  $\text{ack}$  or  $\overline{\text{ack}}$ .
- In iteration  $i$ , Answer Co-training uses the Acknowledgment classifiers,  $\text{ack} H_{i-1}^{i-1}$  of the  $i-1^{\text{th}}$  iteration, to induce more features (detailed in Section 5.2),  $\text{ack} \Phi^i$  which is then concatenated to its original feature vector,  $\text{ans} \Phi^i$ , to get the new feature vector  $(\text{ans} \Phi^i, \text{ack} \Phi^i)$  (we use  $(\vec{A}, \vec{B})$  to represent concatenation of two vectors, where the length of the new vector is  $|\vec{A}| + |\vec{B}|$ ). The concatenated feature vector is then used to train the answer classifiers  $\text{ans} S^i$  of the  $i^{\text{th}}$  iteration.
- The STRUCT view of Answer Co-training uses predictions from the acknowledgment classifier that was trained on the STRUCT view of Acknowledgment Co-training, and similarly for the PATTERN view, so that the concatenated features vectors still remain independent.

## 5.2 Inducing Features from Acknowledgments

Given a thread and acknowledgment tags on some of the posts, the most obvious feature that can be induced on an answer post is a `hasAck` feature which is `True` if any child of this post is marked as an acknowledgment; else `False`. All features that we generated are listed in Table 2.

ACK Feature	Description
Has Ack	True if this post has a reply that is tagged as an acknowledgment; else False.
Ack Distance	The number of posts, in the chronological order, between this post and its acknowledgment.
Last Ack Distance	The number of posts, in the chronological order, between this post and the last acknowledgment post in the thread.

Table 2: Features induced from Acknowledgments

## 6 Experiments

We crawled about 140K threads from Apple Discussions<sup>4</sup>. From these, after discarding those with no replies, 303 threads were randomly chosen, and manually tagged. The inter-annotator agreement<sup>5</sup> between 3 annotators for this task was 0.71. For the experiments, the training set had 3 of the tagged threads and the remaining 300 formed the test set, the statistics of which are in Table 3.

Statistics	Training Set	Test Set
No. of Threads	3	300
Avg. Length of Threads	6.3	5.8
Avg. Answers per Thread	1.9	1.8
Fraction of Answers with Question Author’s reply <sup>6</sup>	47.4%	30.9%

Table 3: Statistics of the Training and Test Sets

We used Support Vector Machines (Vapnik, 1995) (implementation from the LibSVM<sup>7</sup> library) for all the individual classifiers,  $\text{ans} S^i$  and

<sup>4</sup><https://discussions.apple.com>

<sup>5</sup>[http://en.wikipedia.org/wiki/Cohen's\\_kappa](http://en.wikipedia.org/wiki/Cohen's_kappa)

<sup>7</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

$ack H^i$ , used in the different views of the co-training instances of ANS CT and ANS-ACK PCT models.

### 6.1 Study of Improvement in Answer Extraction Accuracy

	STRUCT	PATTERN	COMBINED
SVM	1.1%	2.5%	28.6%
ANS CT	55.6%	55.6%	55.6%
ANS-ACK PCT	63.7%	63.6%	67.8%

Table 4: F Scores for the Answer Extraction task

To demonstrate the benefits of co-training, we first trained a supervised classifier (SVM) on the training set for answer extraction, separately on the two views – STRUCT and PATTERN. With such a little amount of training data, the classifiers gave unimpressive F scores (van Rijsbergen, 1979), shown in the first row of Table 4. The COMBINED classifier is a combination of the individual STRUCT and PATTERN classifiers, computed as:  $(P(\text{ans}|S_{combined}) \propto P(\text{ans}|S_{struct}) \times P(\text{ans}|S_{pattern}))$ ; and similarly for  $\overline{\text{ans}}$ . The post is tagged as  $\text{ans}$  if  $P(\text{ans}|S_{combined}) \geq P(\overline{\text{ans}}|S_{combined})$ . Else, it is  $\overline{\text{ans}}$ .

Next, we performed 40 iterations of co-training and in each step, 5 threads with the most confident predictions were added by each view from the unlabeled pool to the training set. If more than one thread had the same confidence, any one thread was chosen randomly. The accuracies achieved by ANS CT after the final iteration (averaged over 3 runs) is listed in Table 4. Clearly, both STRUCT and PATTERN classifiers drastically improved their F scores and the COMBINED classifier showed a substantial 94% improvement over the SVM baseline. The growth of F score of the two sub-classifiers as the co-training proceeds, is plotted in Figure 4. It can be seen that both the classifiers reached their best within 10 iterations and did not improve any further.

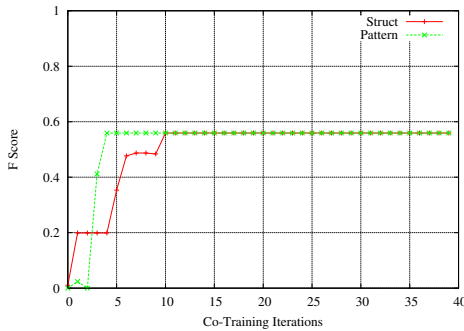


Figure 4: ANS CT: F-scores of the STRUCT and PATTERN sub-classifiers after each iteration

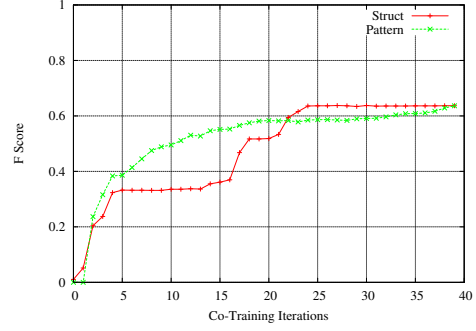


Figure 5: ANS-ACK PCT: F scores of the STRUCT and PATTERN sub-classifiers of the *Answer Classifier* after each iteration

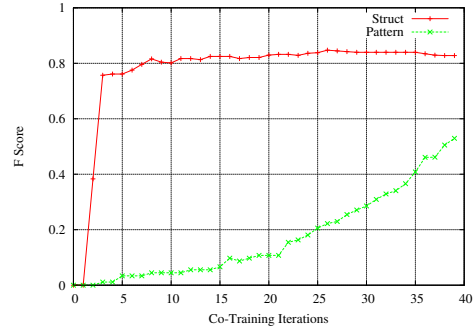


Figure 6: ANS-ACK PCT: F scores of the STRUCT and PATTERN sub-classifiers of the *Acknowledgment Classifier* after each iteration

For ANS-ACK PCT, we performed 40 co-training iterations similar to ANS CT; the number of threads chosen after each iteration was similarly set to 5, for both answer and acknowledgment instances. The answer classification accuracies achieved by ANS-ACK PCT after the final iteration (averaged over 3 runs) is also listed in Table 4. Similar to ANS CT, both STRUCT and PATTERN classifiers improved their F scores significantly. The COMBINED classifier showed a substantial 137% improvement over the SVM baseline and 22% improvement over ANS CT. The F score growth of the answer and the acknowledgment classifiers as the co-training proceeds, is plotted in Figure 5 and Figure 6 respectively. Unlike Figure 4, the answer classifiers in Figure 5 continue to improve even after 10 iterations, since the acknowledgment instance is supplying it with more signals. In Figure 6, the PATTERN sub-classifier of the acknowledgment classifier constantly improved throughout the 40 iterations even though its STRUCT counterpart stabilized in about 10 iterations. The F scores of the acknowledgment classifiers at the end of the final iteration was 82.8%, 52.9% and 81.9% respectively for STRUCT, PATTERN and COMBINED.

## 6.2 Comparison between approaches

	Precision	Recall	F score
SVM - STRUCT	75.0%	0.5%	1.1%
SVM - PATTERN	25.9%	1.3%	2.5%
ANS CT	40.6%	88.0%	55.6%
ANS-ACK PCT	56.8%	84.1%	67.8%
CONG	29.7%	55.6%	38.7%

Table 5: Comparison of accuracy measures of different methods for the answer classification task

Graph Propagation based Answer Extraction by Cong et al. (Cong et al., 2008) is an unsupervised method for extracting answers from discussion forums. It is based on the premise that a correct answer will be repeated often within the discussion and thus, the similarity of the post to other posts can be used as a measure of their “answer-ness”. We used our own implementation of this algorithm, referred to as CONG in the experiments. The similarity between posts is computed using Kullback-Leibler divergence (Kullback, 1997), which was reported by the authors to have given the best performance. The Precision, Recall and F scores of CONG are compared with those of the proposed methods in Table 5.

Table 5 shows that both proposed methods perform substantially better than CONG: ANS CT exceeds it by 43.6% and ANS-ACK PCT surpasses it by 75.1% (F score). Consequently, we can conclude that with very small training data, it is possible to achieve high accuracies compared to extracting them in an unsupervised manner.

### 6.3 Discussion: Answer and Acknowledgment Patterns

This section studies the patterns that were mined from answers and acknowledgments, which reveal the types of sentence structures that frequently appear in them. Some of the interesting answer patterns are listed in Table 6. They are grouped into Imperative, Factual, Conditional and Questions, based on manual inspection. Similarly, the acknowledgment sentences also showed interesting patterns, manually grouped into Action and Others, listed in Table 7. From inspecting the answer and acknowledgment patterns, we conjecture that it should be possible to build classifiers based on rules defined over the structure of the sentence, without requiring access to a training set.

## 7 Conclusion and Future Work

In this paper, we proposed two semi-supervised methods for extracting answers from discussion

Pattern Type	Examples
Imperative Sentence	1. go - to - NNS - NN - on 2. you - can - VB - NN 3. VBG - your - NN - NN 4. VB - to - NNS - NN 5. VBG - your - NN 6. check - NN - NN
Fact	7. is - VBZ - not - NN
Conditional Statement	8. if - you - VBP - NN
Questions	9. have - VB - you - tri - VBN - VBG - NN 10. have - you - VBN - VBG - NN

Table 6: Mined Answer Patterns

Pattern Type	Examples
Action	1. i - VBP - to - VB 2. i - VBP - NN - it 3. i - not - VB - NN 4. i - have - VBP - NN, 5. i - am - VBP - VBG - NN
Others	6. but - i - VBP 7. i - am - VBP - sure

Table 7: Mined Acknowledgment Patterns

threads. We showed how the structural features and sentence construction patterns could be engineered into a co-training setting such that by using a very small training set, and the large amount of unlabeled data available, answers could be extracted with high accuracy, substantially surpassing that attained by an unsupervised method. To demonstrate the benefits of our method, we also showed that completely supervised methods would fail to train a decent model with the very little training data that we used.

In one of our methods, we motivated and introduced a related task of identifying acknowledgments to the answers, which was cast in a parallel co-training setting. We proposed new features which the answer labeling instance could induce from the acknowledgment instance. Our experiments showed that having access to this view of the discussion thread substantially improved the answer extraction accuracy.

Our work opens up new directions of research. In the parallel co-training setting, other than inducing features, the co-training instances are essentially independent. In future, we plan to extend it such that the two instances would collaboratively label new threads; this should lead to higher gains since the instances would now strive to achieve higher coherence between their labels. Also, extending the method to extract answers at a lower granularity like a snippet or a sentence, instead of at a post level would be advantageous for domains that have more factoid type answers.

## References

- A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proc. eleventh annual conference on Computational learning theory, COLT' 98*, pages 92–100.
- C. Callison-Burch and M. Osborne. 2003. Bootstrapping parallel corpora. In *Proc. HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond - Volume 3, HLT-NAACL-PARALLEL '03*, pages 44–49. Association for Computational Linguistics.
- C. Callison-Burch. 2002. Co-training for statistical machine translation. In *Proc. of the 6th Annual CLUK Research Colloquium*.
- R. Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75, July.
- R. Catherine, A. Singh, R. Gangadharaiah, D. Raghu, and K. Visweswariah. 2012. Does similarity matter? the case of answer extraction from technical discussion forums. In *Proc. 24th International Conference on Computational Linguistics, COLING*, pages 175–184.
- G. Cong, L. Wang, C. Lin, Y. Song, and Y. Sun. 2008. Finding question-answer pairs from online forums. In *The 31st Annual International ACM SIGIR Conference*, pages 467–474.
- S. Ding, G. Cong, C. Y. Lin, and X. Zhu. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. In *Meeting of the Association for Computational Linguistics (ACL)*.
- M. Elsner and E. Charniak. 2010. Disentangling chat. *Computational Linguistics*, 36:389–409.
- A. Gandhe, D. Raghu, and R. Catherine. 2012. Domain adaptive answer extraction for discussion boards. In *Proc. 21st international conference companion on World Wide Web, WWW '12 Companion*, pages 501–502. ACM.
- R. Gangadharaiah and R. Catherine. 2012. Prism: discovering and prioritizing severe technical issues from product discussion forums. In *Proc. 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 1627–1631.
- L. Hong and B. D. Davison. 2009. A classification-based approach to question answering in discussion boards. In *Proc. 32nd Annual Intl ACM SIGIR Conf. on Research and Dev. in Information Retrieval*.
- J. Huang, M. Zhou, and D. Yang. 2007. Extracting chatbot knowledge from online discussion forums. In *Proc. 20th international joint conference on Artificial intelligence, IJ-CAI'07*, pages 423–428.
- V. Jijkoun and M. de Rijke. 2005. Retrieving answers from frequently asked questions pages on the web. In *Proc. 14th ACM international conference on Information and knowledge management, CIKM '05*, pages 76–83.
- Nitin Jindal and Bing Liu. 2006. Identifying comparative sentences in text documents. In *In Proc. 29th SIGIR*.
- S. N. Kim, L. Wang, and T. Baldwin. 2010. Tagging and linking web forum posts. In *Proc. Fourteenth Conference on Computational Natural Language Learning, CoNLL '10*, pages 192–202. Association for Computational Linguistics.
- S. N. Kim, L. Cavedon, and T. Baldwin. 2012. Classifying dialogue acts in multi-party live chats. In *Proc. 26th Pacific Asia Conference on Language, Information and Computation*, pages 463–472.
- S. Kullback. 1997. *Information Theory and Statistics*. Dover Publications.
- H. Lakkaraju, C. Bhattacharyya, I. Bhattacharya, and S. Merugu. 2011. Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In *Proc. 11th SIAM International Conference on Data Mining, SDM '11*, pages 498–509.
- J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. 17th International Conference on Data Engineering*. IEEE Computer Society.
- M.F. Porter. 1980. An algorithm for suffix stripping. In *Program*.
- P. Raghavan, R. Catherine, S. Ikbal, N. Kambhatla, and D. Majumdar. 2010. Extracting problem and resolution information from online discussion forums. In *Proc. 16th International Conference on Management of Data, COMAD*.
- S. Sarawagi, S. Chakrabarti, and S. Godbole. 2003. Cross-training: learning probabilistic mappings between topics. In *Proc. ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03*, pages 177–186.
- S. Sarencheh, V. Potdar, E. Yeganeh, and N. Firoozeh. 2010. Semi-automatic information extraction from discussion boards with applications for anti-spam technology. In *Computational Science and Its Applications - ICCSA 2010*, volume 6017 of *Lecture Notes in Computer Science*, pages 370–382. Springer Berlin Heidelberg.
- A. Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proc. second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, NAACL '01*.
- J. Seo, B. Croft, and D. A. Smith. 2009. Online community search using thread structure. In *Proceeding of the 18th ACM conference on Information and knowledge management, CIKM '09*, pages 1907–1910. ACM.
- L. Shrestha and K. McKeown. 2004. Detection of question-answer pairs in email conversation. In *Proc. 20th International Conference on Computational Linguistic (COLING)*.
- C. J. van Rijsbergen. 1979. *Information Retrieval (2nd ed.)*. Butterworth.
- V. N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Y. Wang and C. P. Rosé. 2010. Making conversational structure explicit: identification of initiation-response pairs within online discussions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 673–676. Association for Computational Linguistics.
- H. Wang, C. Wang, C. Zhai, and J. Han. 2011. Learning online discussion structures by conditional random fields. In *Proc. 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11*, pages 435–444.
- X. Xue, J. Jeon, and W. B. Croft. 2008. Retrieval models for question and answer archives. In *Proc. 31st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '08*, pages 475–482. ACM.
- W. Yang, Y. Cao, and C. Lin. 2009. A structural support vector method for extracting contexts and answers of questions from online forums. In *Proc. 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, EMNLP '09*, pages 514–523.

# WordTopic-MultiRank : A New Method for Automatic Keyphrase Extraction

Fan Zhang<sup>†</sup> Lian'en Huang<sup>†</sup> Bo Peng<sup>‡</sup>

<sup>†</sup>The Shenzhen Key Lab for Cloud Computing Technology and Applications  
Peking University Shenzhen Graduate School, Shenzhen 518055, P.R.China

fan.zhgf@gmail.com, hle@net.pku.edu.cn

<sup>‡</sup>Institute of Network Computing and Information Systems  
Peking University, Beijing 100871, P.R.China

pb@net.pku.edu.cn

## Abstract

Automatic keyphrase extraction aims to pick out a set of terms as a representation of a document without manual assignment efforts. Supervised and unsupervised graph-based ranking methods have been studied for this task. However, previous methods usually computed importance scores of words under the assumption of single relation between words. In this work, we propose WordTopic-MultiRank as a new method for keyphrase extraction, based on the idea that words relate with each other via multiple relations. First we treat various latent topics in documents as heterogeneous relations between words and construct a multi-relational word network. Then, a novel ranking algorithm, named Biased-MultiRank, is applied to score the importance of words and topics simultaneously, as words and topics are considered to have mutual influence on each other. Experimental results on two different data sets show the outstanding performance and robustness of our proposed approach in automatic keyphrase extraction task.

## 1 Introduction

Keyphrases refer to the meaningful words and phrases that can precisely and compactly represent documents. Appropriate keyphrases help users a lot in better grasping and remembering key ideas of articles, as well as fast browsing and reading. Moreover, qualities of some information retrieval and natural language processing tasks have been improved with the help of document keyphrases, such as document indexing, categorizing, cluster-

ing and summarizing (Gutwin et al., 1999; Krulwich and Burkey, 1996; Hammouda et al., 2005).

Usually, keyphrases are manually assigned by authors, which is time consuming. With the fast development of Internet, it becomes impractical to label them by human effort as articles on the Web increase exponentially. Therefore, automatic keyphrase extraction plays an important role in keyphrases assignment task.

In most existing work, words are assumed under a single relation and then scored or judged within it. Considering the famous TextRank (Mihalcea and Tarau, 2004), a term graph under a single relatedness was built first, then a graph-based ranking algorithm, such as PageRank (Page et al., 1999), was used to determine the importance score for each term. Another compelling example is (Liu et al., 2010), where words were scored under each topic separately.

In this study, inspired by some multi-relational data mining techniques, such as (Ng et al., 2011), we assume each topic as a single relation type and construct an intra-topic word network for each relation type. In other words, it is to map word relatedness within multiple topics to heterogeneous relations, meaning that words have interactions with others based on different topics.

A multi-relational words example of our proposed WordTopic-MultiRank model is shown in Figure 1(a). There are four words and three relations in this example, implying that there are three potential topics contained in the document. Further, we represent such multi-relational data in a tensor shape in Figure 1(b), where each two-dimensional plane represents an adjacency matrix for one type of topics. Then the heterogeneous network can be depicted as a tensor of size  $4 \times 4 \times 3$ , where  $(i, j, k)$  entry is nonzero if the  $i$ th word is related to the  $j$ th word under  $k$ th topic.

After that, we raise a novel measurement of word relatedness considering different topics, and

<sup>‡</sup>Corresponding author.



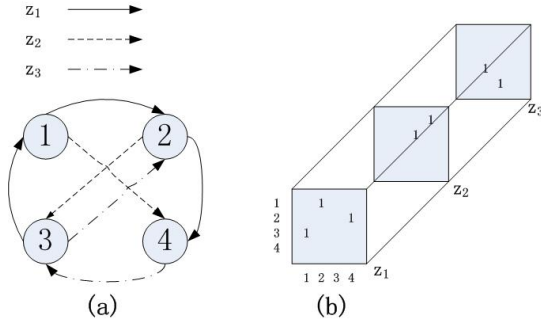


Figure 1: (a) An example of multi-relational words in graph representation and (b) the corresponding tensor representation.

then apply Biased-MultiRank algorithm to deal with multi-relational words for co-ranking purpose, based on the idea that words and topics have mutual influence on each other. More specifically, a word, connected with highly scored words via highly scored topics, should receive a high score itself, and similarly, a topic, connecting highly scored words, should get a high score as well.

Experiments have been performed on two different data sets. One is a collection of scientific publication abstracts, while the other consists of news articles with human-annotated keyphrases. Experimental results demonstrate that our WordTopic-MultiRank method outperforms representative baseline approaches in specified evaluation metrics. And we have investigated how different parameter values influence the performance of our method.

The rest of this paper is organized as follows. Section 2 introduces related work. In Section 3, details of constructing and applying WordTopic-MultiRank model are presented. Section 4 shows experiments and results on two different data sets. Finally, in Section 5, conclusion and future work are discussed.

## 2 Related Work

Existing methods for keyphrase extraction task can be divided into supervised and unsupervised approaches. The supervised methods mainly treat keyphrase extraction as a classification task, so a model needs to be trained before classifying whether a candidate phrase is a keyphrase or not. Turney (1999) firstly utilized a genetic algorithm with parameterized heuristic rules for keyphrase extraction, then Hulth (2003) added more linguistic knowledge as features to achieve better perfor-

mance. Jiang et al. (2009) employed linear Ranking SVM, a learning to rank method, to extract keyphrase lately. However, supervised methods require a training set which would demand time-consuming human-assigned work, making it impractical in the vast Internet space. In this work, we principally concentrate on unsupervised methods.

Among those unsupervised approaches, clustering and graph-based ranking methods showed good performance in this task. Representative studies of clustering approaches are (Liu et al., 2009) and (Grineva et al., 2009). Liu et al. (2009) made use of clustering methods to find exemplar terms and then selected terms from each cluster as keyphrases. Grineva et al. (2009) applied graph community detection techniques to partition the term graph into thematically cohesive groups and selected groups that contained key terms, discarding groups with unimportant terms. But as is widely known, one of the major difficulties in clustering is to predefine the cluster number which influences performance heavily.

As for basic graph-based approaches, such as (Mihalcea and Tarau, 2004) and (Litvak and Last, 2008), a graph based on word linkage or word similarity was first constructed, then a ranking algorithm was used to determine the importance score of each term. Wan et al. (2007) presented an idea of extracting summary and keywords simultaneously under the assumption that summary and keywords of the same document can be mutually boosted. Moreover, Wan and Xiao (2008a) used a small number of nearest neighbor documents for providing more knowledge to improve performance and similarly, Wan and Xiao (2008b) made use of multiple documents with a cluster context. Recently, topical information was under consideration to be combined with graph-based approaches. One of the outstanding studies was Topic-sensitive PageRank (Haveliwala, 2002), which computed scores of web pages by incorporating topics of the context. As another representative, Topical PageRank (Liu et al., 2010) applied a Biased PageRank to assign an importance score to each term under every latent topic separately.

To the best of our knowledge, previous graph-based researches are based on the assumption that all words exist under a unified relation, while in this work, we view latent topics within documents



as word relations and words as multi-relational data, in order to make full use of word-word relatedness, word-topic interaction and inter-topic impacts.

### 3 WordTopic-MultiRank Method

In this section, we will introduce our proposed WordTopic-MultiRank method in details, including topic decomposition, word relatedness measurement, heterogeneous network construction and Biased-MultiRank algorithm.

#### 3.1 Topic Detection via Latent Dirichlet Allocation

There are some existing methods to infer latent topics of words and documents. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is adopted in our work as it is more feasible for inference and it can reduce the risk of over-fitting.

Firstly, we denote the learning corpus for LDA as  $C$ , and  $|C|$  represents the total number of documents in  $C$ . The  $i$ th document in the corpus is denoted as  $d_i$ , in which  $i = 1, 2, \dots, |C|$ . Then, words are denoted as  $w_{ij}$  where  $i$  indicates that word  $w_{ij}$  appears in document  $d_i$  and  $j$  refers to  $j$ th position in  $d_i$  ( $j = 1, 2, \dots, |d_i|$ ,  $|d_i|$  is the total word number in  $d_i$ ). Further, topics inferred from  $|C|$  is  $z_k$ ,  $k = 1, 2, \dots, |T|$ , while  $T$  stands for the topic set detected from  $C$  and  $|T|$  is the total number of topics.

According to LDA, observed words in each document are supposed to be generated by a document-specific mixture of corpus-wide latent topics. More specifically, each word  $w_{ij}$  in document  $d_i$  is generated by first sampling a topic  $z_k$  from  $d_i$ 's document-topic multinomial distribution  $\theta_{d_i}$ , and then sampling a word from  $z_k$ 's topic-word multinomial distribution  $\phi_{z_k}$ . And each  $\theta_{d_i}$  is generated by a conjugate Dirichlet prior with parameter  $\alpha$ , while each  $\phi_{z_k}$  is generated by a conjugate Dirichlet prior with parameter  $\beta$ . The full generative model for  $w_{ij}$  is given by:

$$p(w_{ij}|d_i, \alpha, \beta) = \sum_{k=1}^{|T|} p(w_{ij}|z_k, \beta)p(z_k|d_i, \alpha) \quad (1)$$

Using LDA, we finally obtain the document-topic distribution, namely  $p(z_k|d_i)$  for all the topics  $z_k$  on each document  $d_i$ , as well as the topic-word distribution, namely  $p(w_{ij}|z_k)$  for all the words  $w_{ij}$  on each topic  $z_k$ .

In this work, we use GibbsLDA++<sup>1</sup>, a C/C++ implementation of LDA using Gibbs Sampling, to detect latent topics.

#### 3.2 Measurement of Word Relatedness under Multi-relations

Next, we apply Bayes' theorem to get word-topic distribution  $p(z_k|w_{ij})$  for every word in a given document  $d_i$ :

$$p(z_k|w_{ij}) = \frac{p(w_{ij}|z_k, \beta)p(z_k|d_i, \alpha)}{\sum_{k=1}^{|T|} p(w_{ij}|z_k, \beta)p(z_k|d_i, \alpha)} \quad (2)$$

Therefore, we can obtain word relatedness as follows:

$$p(w_{im}|w_{in}, z_k) = p(w_{im}|z_k)p(z_k|w_{in}) \quad (3)$$

where  $m, n = 1, 2, \dots, |d_i|$ , and  $p(w_{im}|w_{in}, z_k)$  represents the relatedness of word  $w_{im}$  and word  $w_{in}$  under  $k$ th topic.

From the view of probability,  $p(z_k|w_{in})$  is the probability of word  $w_{in}$  being assigned to topic  $z_k$  and  $p(w_{im}|z_k)$  is the probability of generating word  $w_{im}$  from the same topic  $z_k$ . Therefore,  $p(w_{im}|w_{in}, z_k)$  shows the probability of generating word  $w_{im}$  if we have observed word  $w_{in}$  under topic  $z_k$ . Obviously, this point of view corresponds with LDA and it connects words via topics.

#### 3.3 Constructing a Heterogeneous Network on Words

Like Figure 1(a) shown in Introduction, now we construct a multi-relational network for words. In the same way mentioned by typical graph-based methods, for every document  $d_i$  in corpus  $C$ , we treat every single word as a vertex and make use of word co-occurrences to construct a word graph as it indicates the cohesion relationship between words in the context of document  $d_i$ . In this process, a sliding window with maximum  $W$  words is used upon the word sequences of documents. Those words appearing in the same window will have a link to each other under all the relations in the network.

Further, we obtain the word relatedness under every topic from Formula (3), and use them as weights of edges for constructing the heterogeneous network. For instance,  $p(w_{im}|w_{in}, z_k)$  is regarded as the weight of the edge from  $w_{in}$  to  $w_{im}$  under  $k$ th relation if there is a co-occurrence relation between the two words in document  $d_i$ .

<sup>1</sup>GibbsLDA++: <http://gibbslda.sourceforge.net>

As (Hulth, 2003) pointed out, most manually assigned keyphrases were noun groups whose pattern was zero or more adjectives followed by one or more nouns. We only take adjectives and nouns into consideration while constructing networks in experiments.

### 3.4 Ranking Algorithm

In our proposed method, we employ Biased-MultiRank algorithm for co-ranking the importance of words and topics. It is obtained by adding prior knowledge of words and topics to Basic-MultiRank, a basic co-ranking scheme designed for objects and relations in multi-relational data. Therefore, we will demonstrate Basic-MultiRank first, then derive Biased-MultiRank algorithm from it.

#### 3.4.1 Basic-MultiRank Algorithm

In this subsection, we take document  $d_i$  into discussion for convenience. First, we call  $\mathcal{A} = (a_{w_{im}, w_{in}, z_k})$  a real  $(2, 1)$ th order  $(|d_i| \times |T|)$ -dimensional rectangular tensor, where  $a_{w_{im}, w_{in}, z_k}$  denotes  $p(w_{im}|w_{in}, z_k)$  obtained in last subsection, in which  $m, n = 1, 2, \dots, |d_i|$  and  $k = 1, 2, \dots, |T|$ . For example, Figure 1(b) is a  $(2, 1)$ th order  $(4 \times 3)$ -dimensional tensor representation of a document, in which there are 4 words and 3 topics.

Then two transition probability tensors  $\mathcal{O} = (o_{w_{im}, w_{in}, z_k})$  and  $\mathcal{R} = (r_{w_{im}, w_{in}, z_k})$  are constructed with respect to words and topics by normalizing all the entries of  $\mathcal{A}$ :

$$o_{w_{im}, w_{in}, z_k} = \frac{a_{w_{im}, w_{in}, z_k}}{\sum_{m=1}^{|d_i|} a_{w_{im}, w_{in}, z_k}} \quad (4)$$

$$r_{w_{im}, w_{in}, z_k} = \frac{a_{w_{im}, w_{in}, z_k}}{\sum_{k=1}^{|T|} a_{w_{im}, w_{in}, z_k}} \quad (5)$$

Here we deal with dangling node problem in the same way as PageRank (Page et al., 1999). Namely, if  $a_{w_{im}, w_{in}, z_k}$  is equal to 0 for all words  $w_{im}$ , which means that word  $w_{in}$  had no link out to any other words via topic  $z_k$ , we set  $o_{w_{im}, w_{in}, z_k}$  to be  $1/|d_i|$ . Likewise, if  $a_{w_{im}, w_{in}, z_k}$  is equal to 0 for all  $z_k$ , which means that word  $w_{in}$  had no link out to words  $w_{im}$  via all topics, we set  $r_{w_{im}, w_{in}, z_k}$  to be  $1/|T|$ . In this way, we ensure that

$$0 \leq o_{w_{im}, w_{in}, z_k} \leq 1, \sum_{m=1}^{|d_i|} o_{w_{im}, w_{in}, z_k} = 1$$

$$0 \leq r_{w_{im}, w_{in}, z_k} \leq 1, \sum_{k=1}^{|T|} r_{w_{im}, w_{in}, z_k} = 1$$

Following the rule of Markov chain, we derive the probabilities like:

$$P[X_t = w_{im}] = \sum_{n=1}^{|d_i|} \sum_{k=1}^{|T|} o_{w_{im}, w_{in}, z_k} \times P[X_{t-1} = w_{in}, Y_t = z_k] \quad (6)$$

$$P[Y_t = z_k] = \sum_{m=1}^{|d_i|} \sum_{n=1}^{|d_i|} r_{w_{im}, w_{in}, z_k} \times P[X_t = w_{im}, X_{t-1} = w_{in}] \quad (7)$$

where subscript  $t$  denotes the iteration number.

Notice that Formula (6) and (7) accord with our basic idea that, a word connected with high probability words via high probability relations, should have a high probability so that it will be visited more likely, and a topic connecting words with high probabilities, should also get a high one.

After employing a product form of individual probability distributions, we decouple the two joint probability distributions in Formula (6) and (7) as follows:

$$P[X_{t-1} = w_{in}, Y_t = z_k] = P[X_{t-1} = w_{in}] P[Y_t = z_k] \quad (8)$$

$$P[X_t = w_{im}, X_{t-1} = w_{in}] = P[X_t = w_{im}] P[X_{t-1} = w_{in}] \quad (9)$$

Considering stationary distributions of words and topics, while  $t$  goes infinity, the WordTopic-MultiRank values are given by:

$$\bar{\mathbf{x}} = [\bar{x}_{w_{i1}}, \bar{x}_{w_{i2}}, \dots, \bar{x}_{w_{i|d_i|}}]^T \quad (10)$$

$$\bar{\mathbf{y}} = [\bar{y}_{z_1}, \bar{y}_{z_2}, \dots, \bar{y}_{z_{|T|}}]^T \quad (11)$$

with

$$\bar{x}_{w_{im}} = \lim_{t \rightarrow \infty} P[X_t = w_{im}] \quad (12)$$

$$\bar{y}_{z_k} = \lim_{t \rightarrow \infty} P[Y_t = z_k] \quad (13)$$

Under the assumptions from Formula (8) to (13), we can derive these from Formula (6) and (7):

$$\bar{x}_{w_{im}} = \sum_{n=1}^{|d_i|} \sum_{k=1}^{|T|} o_{w_{im}, w_{in}, z_k} \bar{x}_{w_{in}} \bar{y}_{z_k} \quad (14)$$

$$\bar{y}_{z_k} = \sum_{m=1}^{|d_i|} \sum_{n=1}^{|d_i|} r_{w_{im}, w_{in}, z_k} \bar{x}_{w_{im}} \bar{x}_{w_{in}} \quad (15)$$

which mean that the score of  $w_{im}$  depends on its weighted-links with other words via all topics and the score of  $z_k$  depends on scores of the words which it connects with.

Now we are able to solve two tensor equations shown below to obtain the WordTopic-MultiRank values of words and relations according to tensor operations Formula (14) and (15):

$$\mathcal{O}\bar{\mathbf{x}}\bar{\mathbf{y}}=\bar{\mathbf{x}} \quad (16)$$

$$\mathcal{R}\bar{\mathbf{x}}^2=\bar{\mathbf{y}} \quad (17)$$

Ng et al. (2011) show the existence and uniqueness of stationary probability distributions  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{y}}$ , then propose MultiRank, an iterative algorithm, to solve Formula (16) and (17) utilizing Formula (14) and (15). We refer it as Basic-MultiRank algorithm, shown as **Algorithm 1**, for the reason that it will be modified later in the following subsection.

---

#### Algorithm 1 Basic-MultiRank algorithm

**Require:** Tensor  $\mathcal{A}$ , initial probability distributions  $\bar{\mathbf{x}}_0$  and  $\bar{\mathbf{y}}_0$  ( $\sum_{m=1}^{|d_i|}[\bar{\mathbf{x}}_0]_{w_m}=1$  and  $\sum_{k=1}^{|T|}[\bar{\mathbf{y}}_0]_{z_k}=1$ ), tolerance  $\epsilon$

**Ensure:** Two stationary probability distributions  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{y}}$

- 1: compute tensor  $\mathcal{O}$  and  $\mathcal{R}$ ;
  - 2: set  $t = 1$ ;
  - 3: Compute  $\bar{\mathbf{x}}_t = \mathcal{O}\bar{\mathbf{x}}_{t-1}\bar{\mathbf{y}}_{t-1}$ ;
  - 4: Compute  $\bar{\mathbf{y}}_t = \mathcal{R}\bar{\mathbf{x}}_t^2$ ;
  - 5: if  $\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1}\| + \|\bar{\mathbf{y}}_t - \bar{\mathbf{y}}_{t-1}\| < \epsilon$ , then stop, otherwise set  $t = t + 1$  and goto Step 3;
  - 6: **return**  $\bar{\mathbf{x}}_t$  and  $\bar{\mathbf{y}}_t$ .
- 

### 3.4.2 Biased-MultiRank Algorithm

Inspired by the idea of Biased PageRank (Liu et al., 2010), we treat document-word distribution  $p(w_{ij}|d_i)$ , which can be computed from Formula (1), and document-topic distribution  $p(z_k|d_i)$ , acquired from topic decomposition, as prior knowledge for words and topics in each document  $d_i$ . Therefore, we modify Formula (16) and (17) by adding prior knowledge to it as follows:

$$(1-\lambda)\mathcal{O}\bar{\mathbf{x}}\bar{\mathbf{y}}+\lambda\bar{\mathbf{x}}_p=\bar{\mathbf{x}} \quad (18)$$

$$(1-\gamma)\mathcal{R}\bar{\mathbf{x}}^2+\gamma\bar{\mathbf{y}}_p=\bar{\mathbf{y}} \quad (19)$$

where,  $\bar{\mathbf{x}}_p=[p(w_{i1}|d_i),p(w_{i2}|d_i),\dots,p(w_{i|d_i}|d_i)]^T$  and  $\bar{\mathbf{y}}_p=[p(z_1|d_i),p(z_2|d_i),\dots,p(z_{|T|}|d_i)]^T$ .

Then we propose Biased-MultiRank, shown as **Algorithm 2**, as a new algorithm to solve the prior-tensors and Formula (18) and (19). Finally it is used in our WordTopic-MultiRank model.

---

#### Algorithm 2 Biased-MultiRank algorithm

**Require:** Tensor  $\mathcal{A}$ , initial probability distributions  $\bar{\mathbf{x}}_0$  and  $\bar{\mathbf{y}}_0$  ( $\sum_{m=1}^{|d_i|}[\bar{\mathbf{x}}_0]_{w_m}=1$  and  $\sum_{k=1}^{|T|}[\bar{\mathbf{y}}_0]_{z_k}=1$ ), prior distribution of words  $\bar{\mathbf{x}}_p$  and topics  $\bar{\mathbf{y}}_p$ , parameters  $\lambda$  and  $\gamma$  ( $0 \leq \lambda, \gamma < 1$ ), tolerance  $\epsilon$

**Ensure:** Two stationary probability distributions  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{y}}$

- 1: compute tensors  $\mathcal{O}$  and  $\mathcal{R}$ ;
  - 2: set  $t = 1$ ;
  - 3: Compute  $\bar{\mathbf{x}}_t = (1-\lambda)\mathcal{O}\bar{\mathbf{x}}_{t-1}\bar{\mathbf{y}}_{t-1} + \lambda\bar{\mathbf{x}}_p$ ;
  - 4: Compute  $\bar{\mathbf{y}}_t = (1-\gamma)\mathcal{R}\bar{\mathbf{x}}_t^2 + \gamma\bar{\mathbf{y}}_p$ ;
  - 5: if  $\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1}\| + \|\bar{\mathbf{y}}_t - \bar{\mathbf{y}}_{t-1}\| < \epsilon$ , then stop, otherwise set  $t = t + 1$  and goto Step 3;
  - 6: **return**  $\bar{\mathbf{x}}_t$  and  $\bar{\mathbf{y}}_t$ .
- 

## 4 Experiment

To evaluate the performance of WordTopic-MultiRank in automatic keyphrase extraction task, we utilize it on two different data sets and describe the experiments specifically in this section.

### 4.1 Experiments on Scientific Abstracts

#### 4.1.1 Data Set

We first employ WordTopic-MultiRank model to conduct experiments on a data set of scientific publication abstracts from the INSPEC database with corresponding manually assigned keyphrases<sup>2</sup>. The data set is also used by Hulth (2003), Mihalcea and Tarau (2004), Liu et al. (2009), and Liu et al. (2010), meaning that it is classically used in the task of keyphrase extraction, and is convenient for comparison.

Actually, this data set contains 2,000 abstracts of research articles and 19,254 manually annotated keyphrases, and is split into 1,000 for training, 500 for validation and 500 for testing.

In this study, we use the 1,000 training documents as corpus  $C$  for topic detection and like other unsupervised ranking methods, 500 test documents are used for comparing the performance with baselines. Following previous work, only the manually assigned uncontrolled keyphrases that occur in the corresponding abstracts are viewed as standard answers.

---

<sup>2</sup>It can be obtained from <http://github.com/snkim/AutomaticKeyphraseExtraction>

#### 4.1.2 Baselines and Evaluation Metrics

We choose methods proposed by Hulth (2003), Mihalcea and Tarau (2004), Liu et al. (2009), and Liu et al. (2010) as baselines for the reason that they are either classical or outstanding in keyphrase extraction task.

Evaluation metrics are *precision*, *recall*, *F1-measure* shown as follows:

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F1 = \frac{2PR}{P+R} \quad (20)$$

where  $TP$  is the total number of correctly extracted keyphrases,  $FP$  is the number of incorrectly extracted keyphrases, and  $FN$  is the number of those keyphrases which are not extracted.

#### 4.1.3 Data Pre-processing and Configuration

Documents are pre-processed by removing stop words and annotated with POS tags using Stanford Log-Linear Tagger<sup>3</sup>.

Based on the research result of (Hulth, 2003), only adjectives and nouns are used in constructing multi-relational words network for ranking, and keyphrases corresponding with following pattern are considered as candidates:

$$(JJ)*(NN|NNS|NNP)+$$

in which, JJ indicates adjectives while NN, NNS and NNP represent various forms of nouns.

At last, top- $M$  keyphrases, which have highest sum scores of words contained in them, are extracted and compared with standard answers after stemming by Porter stemmer<sup>4</sup>.

In experiments, we set  $\alpha=1$ ,  $\beta=0.01$  for Formula (1) to (3) empirically, and  $\lambda=0.5$ ,  $\gamma=0.9$  for Formula (18), (19) indicated by (Li et al., 2012). Influences of these parameters will not be discussed further in this work as they have been studied intensively in previous researches.

#### 4.1.4 Experimental Results

In this subsection, we investigate how different parameter values influence performance of our proposed model first, then compare the best results obtained by baseline methods and our model.

First of all, we inspect influences of topic number  $|T|$  on our model performance. Table 1 shows experimental results when  $|T|$  ranges from 20 to 100 while setting window size  $W=2$  and max extracted number  $M=10$ .

<sup>3</sup><http://nlp.stanford.edu/software/tagger.shtml>

<sup>4</sup><http://tartarus.org/martin/PorterStemmer/>

Topic Number	Precision	Recall	F1
20	0.463	0.498	0.479
40	0.464	0.500	0.480
60	0.465	0.502	<b>0.482</b>
80	0.462	0.499	0.480
100	0.462	0.499	0.480

Table 1: Influence of Topic Number  $|T|$

From Table 1, we observe that the performance does not change much when the number of topics varies, showing our model’s robustness under the situation that the actual number of topics is unknown, which is commonly seen in Information Retrieval and Natural Language Processing applications. We can see that  $|T|=60$  produces the best result for this corpus, so we choose 60 for  $|T|$  in comparison with baselines.

Then, we fix  $|T|=60$  and  $M=10$  to demonstrate how our model is affected by the windows size  $W$ . Table 2 presents the metrics when  $W$  ranges from 2 to 10.

Window Size	Precision	Recall	F1
2	0.465	0.502	<b>0.482</b>
4	0.461	0.496	0.477
6	0.462	0.500	0.480
8	0.461	0.499	0.479
10	0.461	0.498	0.478

Table 2: Influence of Window Size  $W$

Our results are consistent with the findings reported by Liu et al. (2009) and Liu et al. (2010), indicating that performance usually does not vary much when  $W$  ranges. More details point out that  $W=2$  is the best.

Moreover, we explore the influence of max extracted number  $M$  by setting  $W=2$  and  $|T|=60$ .

$M$	Precision	Recall	F1
5	<b>0.602</b>	0.393	0.475
10	0.465	0.502	<b>0.482</b>
15	0.420	<b>0.550</b>	0.476

Table 3: Influence of Max Extracted Number  $M$

Table 3 indicates that as  $M$  increases, *precision* falls down while *recall* raises up, and  $M=10$  performs best in *F1-measure*.

At last, Table 4 shows the best results of baseline methods and our proposed model. In fac-

Method	Precision	Recall	F1
Hulth’s (Hulth, 2003)	0.252	0.517	0.339
TextRank (Mihalcea and Tarau, 2004)	0.312	0.431	0.362
Topical PageRank (Liu et al., 2010)	0.354	0.183	0.242
Clustering (Liu et al., 2009)	0.350	<b>0.660</b>	0.457
WordTopic-MultiRank	<b>0.465</b>	0.502	<b>0.482</b>

Table 4: Comparison on Scientific Abstracts

Method	Precision	Recall	F1
ExpandRank(Wan and Xiao, 2008a)	0.288	0.354	0.317
CollaRank(Wan and Xiao, 2008b)	0.283	0.348	0.312
Topical PageRank(Liu et al., 2010)	0.282	0.348	0.312
WordTopic-MultiRank	<b>0.296</b>	<b>0.399</b>	<b>0.340</b>

Table 5: Comparison on DUC2001

t, the best result of (Hulth, 2003) was obtained by adding POS tags as features for classification, while running PageRank on an undirected graph, which was built via using window  $W=2$  on word sequence, resulted best of (Mihalcea and Tarau, 2004). According to (Liu et al., 2009), spectral clustering method got best performance in *precision* and *F1-measure*. On the other hand, Topical PageRank (Liu et al., 2010) performed best when setting window size  $W=10$ , topic number  $|T|=1,000$ . Since the influences of parameters have been discussed above, we set  $W=2$ ,  $|T|=60$  and  $M=10$  as they result in best performance of our model on the same data set.

Table 4 demonstrates that our proposed model outperforms all baselines in both *precision* and *F1-measure*. Noting that baseline methods are all under a single relation type assumption for word relatedness, estimations of their word ranking scores are limited, while WordTopic-MultiRank assumes words as multi-relational data and considers interactions between words and topics more comprehensively.

## 4.2 Experiments on DUC2001

In order to show the generalization performance of our model, we also conduct experiments on another data set for automatic keyphrase extraction task and describe it in this subsection briefly.

Following (Wan and Xiao, 2008a), (Wan and Xiao, 2008b) and (Liu et al., 2010), a data set annotated by Wan and Xiao<sup>5</sup> was used in this experiment for evaluation. This data set is the testing part of DUC2001(Over and Yen, 2004), con-

<sup>5</sup><http://wanxiaojun1979.googlepages.com/>

taining 308 news articles with 2,488 keyphrases manually labeled. And at most 10 keyphrases were assigned to each document. Again, we choose *precision*, *recall* and *F1-measure* as evaluation metrics and use the train part of DUC2001 for topic detection. At last, keyphrases extracted by our WordTopic-MultiRank model will be compared with the ones occurring in corresponding articles after stemming.

As indicated in (Wan and Xiao, 2008b), performance on test set does not change much when co-occurrence window size  $W$  ranges from 5 to 20, and (Liu et al., 2010) also reports that it does not change much when topic number ranges from 50 to 1,500. Therefore, we pick co-occurrence window size  $W=10$  and topic number  $|T|=60$  to run WordTopic-MultiRank model. As for Keyphrase number  $M$ , we vary it from 1 to 20 to obtain different performances. Results are shown in Figure 2.

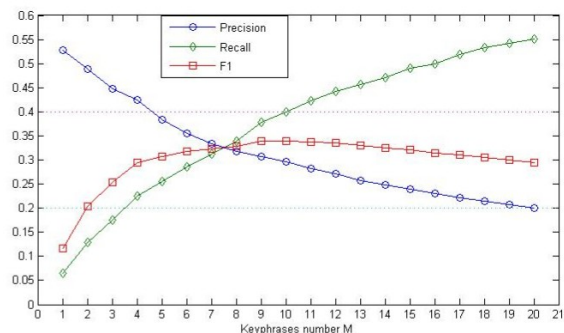


Figure 2: performance vs. Keyphrase number  $M$

From Figure 2, we can observe how performances of our model change with  $M$ . Actually,

as  $M$  increases from 1 to 20, *precision* decreases from 0.528 to 0.201 in our experiment, while *recall* increases from 0.065 to 0.551. As for *F1-measure*, it obtains maximum value 0.340 when  $M=10$  and decreases gradually as  $M$  leaves 10 farther. Therefore,  $W=10$ ,  $|T|=60$  and  $M=10$  are optimal for our proposed method on this test set.

Table 5 lists the best performance comparison between our method and previous ones. All previous methods perform best on DUC2001 test set while setting co-occurrence window size  $W=10$  and Keyphrase number  $M=10$ , which is consistent with our model.

Experimental results on this data set demonstrate the effectiveness of our proposed model again as it outperforms baseline methods over all three metrics.

## 5 Conclusion and Future Work

In this study, we propose a new method named WordTopic-MultiRank for automatic keyphrase extraction task. It treats words in documents as objects and latent topics as relations, assuming words are under multiple relations. Based on the idea that words and topics have mutual influence on each other, our model ranks importance of words and topics simultaneously, then extracts highly scored phrases as keyphrases. In this way, it makes full use of word-word relatedness, word-topic interaction and inter-topic impacts. Experiments demonstrate that WordTopic-MultiRank achieves better performance than baseline methods on two different data sets. It also shows the good effectiveness and strong robustness of our method after we explored the influence of different parameter values.

In future work, for one thing, we would like to investigate how different corpora influence our method and choose a large-scale and general corpus, such as Wikipedia, for experiments. For another, exploring more algorithms to deal with heterogeneous relation network may help to unearth more knowledge between words and topics, and improve our model performance.

## Acknowledgments

This research is financially supported by NSFC Grant 61073082 and NSFC Grant 61272340.

## References

- D.M. Blei, A.Y. Ng, and M.I. Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.
- Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multi-theme documents. In *Proceedings of the 18th international conference on World wide web*, pages 661–670. ACM.
- Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1):81–104.
- Khaled M Hammouda, Diego N Matute, and Mohamed S Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *Machine Learning and Data Mining in Pattern Recognition*, pages 265–274. Springer.
- Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.
- A. Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of EMNLP*, pages 216–223.
- Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 756–757. ACM.
- Bruce Krulwich and Chad Burkey. 1996. Learning user information interests through extraction of semantically significant phrases. In *Proceedings of the AAAI spring symposium on machine learning in information access*, pages 100–112.
- Xutao Li, Michael K Ng, and Yunming Ye. 2012. Har: Hub, authority and relevance scores in multi-relational data for query search. In *SDM*, pages 141–152.
- Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on multi-source multilingual information extraction and summarization*, pages 17–24. Association for Computational Linguistics.
- Z. Liu, P. Li, Y. Zheng, and M. Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of EMNLP*, pages 257–266.
- Z. Liu, W. Huang, Y. Zheng, and M. Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of EMNLP*, pages 366–376.
- R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, pages 404–411.

- M.K.P. Ng, X. Li, and Y. Ye. 2011. Multirank: co-ranking for objects and relations in multi-relational data. In *Proceedings of the 17th ACM SIGKDD*, pages 1217–1225.
- Paul Over and James Yen. 2004. Introduction to duc-2001: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of DUC 2004 Document Understanding Workshop, Boston*.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: bringing order to the web.
- Peter D Turney. 1999. Learning to extract keyphrases from text. national research council. *Institute for Information Technology, Technical Report ERB-1057*.
- X. Wan and J. Xiao. 2008a. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of AAAI*, pages 855–860.
- Xiaojun Wan and Jianguo Xiao. 2008b. Col-labrank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 969–976. Association for Computational Linguistics.
- X. Wan, J. Yang, and J. Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *ACL*, page 552.

# Towards Contextual Healthiness Classification of Food Items - A Linguistic Approach

Michael Wiegand and Dietrich Klakow

Spoken Language Systems

Saarland University

D-66123 Saarbrücken, Germany

{Michael.Wiegand|Dietrich.Klakow}@lsv.uni-saarland.de

## Abstract

We explore the feasibility of contextual healthiness classification of food items. We present a detailed analysis of the linguistic phenomena that need to be taken into consideration for this task based on a specially annotated corpus extracted from web forum entries. For automatic classification, we compare a supervised classifier and rule-based classification. Beyond linguistically motivated features that include sentiment information we also consider the prior healthiness of food items.

## 1 Introduction

Food plays a substantial part in each of our lives. With the growing health awareness in many parts of the population, there is consequently a high demand for the knowledge about healthiness of food. In view of the variety of both different types of food and nutritional aspects it does not come as a surprise that there is no comprehensive repository of that knowledge. Since, however, much of this information is preserved in natural language text, we assume that it is possible to acquire some of this knowledge automatically with the help of natural language processing (NLP).

In this paper, we take a first step towards this endeavour. We try to identify mentions that a food item is healthy (1) or unhealthy (2).

(1) There is not a healthy diet without a lot of fruits, vegetables and salads.

(2) The day already began unhealthy: I had a piece of cake for breakfast.

This task is a pre-requisite of more complex tasks, such as finding food items that are suitable for certain groups of people with a particular health condition (3) or identifying reasons for the healthiness or unhealthiness of particular food items (4).

(3) Vegetables are healthy, in particular, if you suffer from diabetes.

(4) Potatoes are healthy since they are actually low in calories.

The major problem of identifying some *Is-Healthy* or *Is-Unhealthy* relation is that the simple co-occurrence of a food item and the word *healthy* or *unhealthy* is not sufficiently predictive as shown in (5)-(7).

(5) Chocolate is *not* healthy.

(6) *The industry says* chocolate is healthy, but I guess this is just a marketing strategy.

(7) *If* chocolate is healthy, then I will run for the next presidential election.

We describe the contextual phenomena that underlie these cases and provide detailed statistics as to how often they occur in a typical text collection. From this analysis we derive features to be incorporated into a classifier.

Our experiments are carried out on German data. We believe, however, that our findings carry over to other languages since the aspects addressed in this work are (mostly) language universal. For the sake of general accessibility, all examples will be given as English translations.

To the best of our knowledge, this is the first work that addresses the classification of healthiness of food items using NLP.

## 2 Related Work

In the food domain, the most prominent research addresses ontology or thesaurus alignment (van Hage et al., 2010), a task in which concepts from different sources are related to each other. In this context, hyponymy relations (van Hage et al., 2005) and part-whole relations (van Hage et al., 2006) have been explored. More recently, Wiegand et al. (2012a) examined extraction methods for relations involved in customer advice in a supermarket. In Chahuneau et al. (2012), sentiment information has been related to food prices with the help of a large corpus consisting of restaurant menus and reviews.

In the health/medical domain, the majority of research focus on domain-specific relations involving entities, such as genes, proteins and



drugs (Cohen and Hersh, 2005). More recently, the prediction of epidemics (Fisichella et al., 2011; Torii et al., 2011; Diaz-Aviles et al., 2012; Munro et al., 2012) has attracted the attention of the research community. In addition, there has been research on processing health-care claims (Popowich, 2005) and detecting sentiment in health-related texts (Sokolova and Bobicev, 2011).

### 3 The Dataset

In order to generate a dataset for our experiments, we used a crawl of *chefkoch.de*<sup>1</sup> (Wiegand et al., 2012a) consisting of 418,558 webpages of food-related forum entries. *chefkoch.de* is the largest German web portal for food-related issues.

While we are aware of the fact that the healthiness of food items is also discussed in scientific texts we think that the text analysis on social media serves its own purpose. The language in social media is much more accessible to the general population. Moreover, social media can be considered as an exclusive repository of *popular wisdom* containing, for example, home remedies.

#### 3.1 Healthiness Markers & Food Items

As it is impractical for us to manually label the entire web corpus with healthiness information, we extracted for annotation sentences in which there is a healthiness marker and a mention of a food item. By healthiness marker, we understand an expression that conveys the property of being healthy. Apart from the word *healthy* itself, we came up with 17 further common expressions (e.g. *nutritious*, *healthful* or *in good health*). Since the word *healthy* covers more than 95% of the mentions of healthiness markers in our entire corpus, however, we decided to restrict our healthiness marker exclusively to mentions of that expression. Thus, our main focus in this classification task is the contextual disambiguation, i.e. the task to decide whether a specific co-occurrence of the expression *healthy* and some food item denotes a genuine *Is-(Un)Healthy* relation.

The food items for which we extract co-occurrences with the healthiness marker *healthy* (Table 7) will henceforth be referred to as *target food items*. In order to obtain a suitable list of items for our experiments, we manually compiled a list of frequently occurring types of food.

<sup>1</sup>[www.chefkoch.de](http://www.chefkoch.de)

#### 3.2 “Unhealthy” vs. “Not Healthy”

In order to obtain instances that express an *Is-Unhealthy* relation, we exclusively consider negated instances of the *Is-Healthy* relation (8). We also experimented with a dataset with mentions of the word *unhealthy* (paired with our target food items) to extract instances such as (9).

(8) I am convinced that cake is *not healthy*.

(9) I am convinced that cake is *unhealthy*.

Using the same target food items, the *unhealthy*-dataset is, however, less than 14% of the size of the *healthy*-dataset. We also found that instances of the *Is-Unhealthy*-relation are not easier to detect on the *unhealthy*-dataset, since the *unhealthy*-dataset produced much poorer classifiers for detecting *Is-Unhealthy* relations than the *healthy*-dataset using negations as a proxy.

### 4 Annotation

Our final dataset comprises 2,440 instances, where each **instance** consists of a sentence with the co-occurrence of some food item and the word *healthy* accompanied by the two sentences immediately preceding and the two sentences immediately following it.

The dataset was manually annotated by two German native speakers. On 4 target food items (this corresponds to 574 target sentences)<sup>2</sup> we measured an inter-annotation agreement of Cohen’s  $\kappa = 0.7374$  (Landis and Koch, 1977) which should be sufficiently high for our experiments.

The annotators had to choose from a rich set of category labels that particularly divide the negative examples (i.e. those cases in which the co-occurrence of the target food item and *healthy* neither expresses an *Is-Healthy* nor an *Is-Unhealthy* relation) into different categories.

In the following, we describe the different category labels. Their distribution is shown in Table 1.

#### 4.1 Is-Healthy Relation (HLTH)

This class describes instances in which there holds an *Is-Healthy* relation between the mention of *healthy* and the target food item (10).

(10) Potatoes are incredibly healthy, versatile in the kitchen and very tasty.

Table 1 shows that less than 20% of the co-occurrences of the target food item and *healthy* express this relation. This may already indicate that its extraction is difficult.

<sup>2</sup>This is the only part of the dataset which was annotated by both annotators in parallel.

Type	Abbrev.	Frequency	Percentage
Is-Healthy	HLTH	488	20.00
Is-Unhealthy	UNHLTH	171	7.01
OTHER:			
No Relation	NOREL	788	32.30
Restricted Relation	RESTR	312	12.79
Unspecified Intersection	INTERS	198	8.11
Embedding	EMB	157	6.43
Comparison Relation	COMP	121	4.96
Unsupported Claim	CLAIM	87	3.57
Other Sense	SENSE	77	3.16
Irony	IRO	25	1.02
Question	Q	16	0.66

Table 1: Statistics of the different (linguistic) phenomena.

## 4.2 Is-Unhealthy Relation (UNHLTH)

We already stated in §3.2 that we consider negated instances (11) as instances for the *Is-Unhealthy* relation. We have a fairly broad notion of negation, e.g. (12) and (13) will also be assigned to this category. These *partial* negations are at least as frequent as *full* negations (11). However, we assume that the latter are often employed only as a means of being polite even though the speaker’s intention is that of a full negation. The fact that we also observed fewer mentions of *unhealthy* co-occurring with a target food item than negated mentions of *healthy* would be in line with this theory (*unhealthy* is usually perceived to be more intense/blunter than *not healthy*).

- (11) Chocolate is not healthy.  
(12) Chocolate is not very healthy.  
(13) Chocolate is hardly healthy.

## 4.3 Other Relations

Apart from the two target relations, we observe the following other relationships:

### 4.3.1 Restricted Relation (RESTR)

This category describes cases in which the *Is-Healthy* relation holds provided some additional condition is fulfilled. Typical conditions address a special kind of preparing the target food item (14) or make quantitative restrictions as to the amount of the target food item to be consumed (15). As such, one cannot infer from restricted relations to general properties of food items.

- (14) Steamed vegetables are extremely healthy.  
(15) A teaspoon of honey each day has been proven to be quite healthy.

### 4.3.2 Unspecified Intersection (INTERS)

In relation extraction, syntactic relatedness between the candidate entities of a relation is usually

considered an important cue (Zhou et al., 2005; Mintz et al., 2009). In particular, the specific *type* of syntactic relation needs to be considered. If in our task *healthy* is an attributive adjective of the target food item (16), this is not an indication of a genuine *Is-Healthy* relation that we are looking for. With this construction, one usually refers to all those entities that share the two properties (*intersection*) of being the target food item and being healthy. This case is different from both *HLTH* (17) and *RESTR* (18).

- (16) I usually buy the healthy fat.  
(17) Fat is healthy.  
(18) I usually buy the healthy fat, the one that contains a high degree of unsaturated fatty acids.

*HLTH*, typically realized as a predicative adjective (17), requires that this intersection of properties includes the *entire* set of entities representing the target food item. For both *RESTR* and *INTERS*, on the other hand, this intersection only includes a proper subset of the target food item. In addition, *RESTR* provides some (vital) additional information about this subset that allows it to be (easily) identified (e.g. the property of containing a high degree of unsaturated fatty acids in (18)). However, for *INTERS*, no further properties are specified in order to identify it – the information of being healthy is not telling as we actually want to find out how to detect healthy food. As a consequence, instances of type *INTERS* are hardly informative when it comes to answering whether a particular food item is healthy or not. We do not even know how large the proportion of the intersection with regard to the overall amount of the target food item is. It may well be extremely small. That is why in this work, instances of *INTERS* will neither be used as evidence for the healthiness nor the unhealthiness of a particular food item.

### 4.3.3 Comparison Relation (COMP)

If the target food item is compared with another food item with regard to their healthiness status (19) & (20), one cannot conclude anything regarding the *absolute* healthiness of the target food item. This is due to the fact that a comparison assumes healthiness as a (continuous) scale rather than a binary (discrete) property. It determines the positions of the two food items relative to each other on that particular scale.

- (19) Honey is healthier than chocolate. (target food item: *honey*)  
(20) Honey is as healthy as chocolate. (target food item: *honey*)

#### 4.3.4 Unsupported Claim (CLAIM)

In our initial data analysis, we found frequent cases in which the author of a forum entry reports a (controversial) statement regarding the healthiness status of a particular food item. These claims are often used as a means of starting a discussion about that issue (21).

(21) Some people claim that chocolate is healthy. What do you make of it?

If it is not possible to infer from such reported statement that the reported view is shared by the author (and we found that this is true for many reported statements), we tag it as *CLAIM*.

#### 4.3.5 Question (Q)

There may also be cases in which the *Is-(Un)Healthy* relation is embedded in a question (22).

(22) Is chocolate healthy?

#### 4.3.6 Irony (IRO)

Irony (23) is a figure of speech that can frequently be observed in user-generated text (Tsur et al., 2010). With a proportion of less than 1%, this, however, does not apply for the forum entries that comprise our data collection.

(23) Everyone knows that sweets are healthy, in particular, chocolate with its many calories even makes you lose weight.

#### 4.3.7 Embedding (EMB)

In addition to the previous categories *CLAIM* and *IRO*, there exist other ways of embedding the healthiness relation into a context so that the general validity of it is discarded. We introduce a common label for all those other remaining types that include, for instance, *modal embedding* (24) or *irrealis construction* (25).

(24) Honey could be healthy.

(25) If chocolate were healthy, people eating it wouldn't put on so much weight.

#### 4.3.8 Other Sense (SENSE)

Both the target food item and the German healthiness cue *gesund* are (potentially) ambiguous expressions. For instance, *gesund* can be part of several multiword expressions, such as *gesunder Menschenverstand* (engl. *common sense*).

#### 4.3.9 No Relation (NOREL)

While in all previously discussed cases the target food item and *healthy* are somehow related, there are cases in which the co-occurrence is merely coincidental (26).

(26) Tomatoes are very healthy and they can be ideally served on bread. (target food item: *bread*)

On our dataset, this is the most frequent label.

## 5 Feature Design

All features we use are summarized in Table 2 along examples. Apart from bag of words (*word*), we use following features:

### 5.1 Linguistic Features

The linguistic features are mainly derived from our quantitative data analysis in §4. Given the limited space of this paper, we will only point out some special properties.

The first group of (linguistic) features (Table 2) is designed to detect some relationship between target food item and *healthy*. The co-occurrence within the same clause is usually a good predictor. There are three features to establish this property: *clause*, *boundary* and *otherFood*.

We already pointed out in §4.3.2 that not only syntactic relatedness between *healthy* and the target food item as such but also the specific syntactic relation plays a decisive role for this task. The two most common relations are that *healthy* is a predicative adjective (of the target food item), which is usually indicative of *HLTH*, and that *healthy* is an attributive adjective (of the target food item), which is usually indicative of *INTERS* (on our dataset in more than 90% of the instances labeled with *INTERS* this is the case). This is reflected by the two features *predRel* and *attrRel* (and the back-off features *pred* and *attr*). An additional feature *attrFood* captures a special construction in which *healthy* as an attributive adjective actually denotes *HLTH* instead of *INTERS*.

For the conditional healthiness *RESTR* (§ 4.3.1), we found two predominant subcategories of restrictions: restrictions with regard to the quantity with which the target food item should be consumed (*quant*) and references to a specific subtype of the target food item, which we want to capture with a few precise surface patterns (*spec*) and a feature that checks whether the target food item precedes an attributive adjective (*attrNoH*).

Table 2 also contains features to detect various contextual embeddings (*opHolder*, *question*, *irrealis*, *modal* and *irony*). *opHolder* is to detect cases of *CLAIM*. We assume once some opinion holder other than the author of the forum post (i.e. 1st person pronoun) is identified, there is a *CLAIM*.

We also investigate whether *healthiness* correlates with *sentiment*. For instance, if the author promotes the healthiness of some food item, does this also coincide with positive sentiment (e.g.

*tasty, good* etc.)? Our features *positive/negative polar* check for the presence of polar expressions.

## 5.2 Knowledge-based Features using a Healthiness Lexicon

We also incorporate features referring to the prior knowledge of healthiness of food items. We use a lexicon introduced in Wiegand et al. (2012b) which covers approximately 3000 food items, and we refer to it as *healthiness lexicon*. Each food item is specified as being either healthy or unhealthy in that lexicon. The healthiness judgment has been carried out based on the general nutrient content of each food item. A detailed description of the annotation scheme and annotation agreement can be found in Wiegand et al. (2012b).

The specific features derived from that lexical resource are listed in Table 2. They are divided into two groups. *prior* describes the prior healthiness of the target food item. Since our task is to determine the *contextual* healthiness, the usage of such a feature is legitimate. The *contextual* healthiness need not to coincide with the *prior* healthiness. For instance, in (27), *chocolate* is described as a healthy food item even though it is a priori considered unhealthy.

(27) Chocolate is healthy as it’s high in magnesium and provides vitamin E.

We use this knowledge as a baseline. If we cannot exceed the classification performance of *prior* (alone), then acquiring the knowledge of healthiness with the help of NLP is hardly effective.

*priorCont* describes the prior healthiness status of *neighbouring food items* in the given context.

## 6 Rule-based Classification

We also examine rule-based classifiers since they can be built without any training data. Each classifier is defined by a (large) conjunction of linguistic features. Features indicating a class other than the target class are used as negated features in that conjunction. The rule-based classifiers only consider features where a positive or negative correlation towards the target class is (more or less) obvious. Table 3 shows the rule-based classifiers for each of our classes. For *HLTH*, it basically states that *healthy* has to be a predicative adjective of the target food item (*predRel*), and the target food item and *healthy* have to appear within the same clause (or there is no boundary sign between them). After that, a long list of negated features follows: *quant*, *spec* and *attrNoH*, for exam-

HLTH	$\text{predRel} \wedge (\text{clause} \vee \neg\text{boundary}) \wedge \neg\text{quant} \wedge \neg\text{spec} \wedge \neg\text{attrNoH} \wedge \neg\text{negTarget} \wedge \neg\text{negHealth} \wedge \neg\text{comp} \wedge \neg\text{opHolder} \wedge \neg\text{modal} \wedge \neg\text{irrealis} \wedge \neg\text{question} \wedge \neg\text{sense} \wedge \neg\text{weird}$
UNHLTH	$\text{predRel} \wedge (\text{clause} \vee \neg\text{boundary}) \wedge \neg\text{quant} \wedge \neg\text{spec} \wedge \neg\text{attrNoH} \wedge (\text{negTarget} \vee \text{negHealth}) \wedge \neg\text{comp} \wedge \neg\text{opHolder} \wedge \neg\text{modal} \wedge \neg\text{irrealis} \wedge \neg\text{question} \wedge \neg\text{sense} \wedge \neg\text{weird}$

Table 3: Rule-based classifiers based on linguistic features (Table 2).

ple, are negated because they are typical cues for *RESTR*. The remaining features are negated since they are either indicative of *UNHLTH*, *COMP*, *EMB*, *CLAIM*, *SENSE*, *IRO* or *Q*. The classifier for *UNHLTH* only differs from *HLTH* in that either of the negation cues, i.e. *negTarget* or *negHealth*, has to be present.

## 7 Experiments

In this section we present the results on automatic classification.

### 7.1 Classification of Individual Utterances

In this subsection, we evaluate the performance of the different feature sets on sentence-level classification using supervised learning and rule-based classification. We investigate the detection of the two classes *HLTH* (§4.1) and *UNHLTH* (§4.2). Each instance to be classified is a sentence in which there is a co-occurrence of a target food item and a mention of *healthy* along its respective context sentences. The dataset was parsed using the Stanford Parser (Rafferty and Manning, 2008). We carry out a 5-fold cross-validation on our manually labeled dataset. As a supervised classifier, we use Support Vector Machines (*SVM<sup>light</sup>* (Joachims, 1999) with a linear kernel). For each class, we train a binary classifier where positive instances represent the class to be extracted while negative instances are the remaining instances of the entire dataset (§4).

#### 7.1.1 Comparison of Various Feature Sets

Table 4 lists the results for various feature sets that we experimented with. *take-all* is an unsupervised baseline that considers all instances of our dataset as positive instances (of the class which is examined, i.e. *HLTH* or *UNHLTH*). In other words, this baseline indicates how well the mere co-occurrence of *healthy* and the target food item predicts either of our two classes.<sup>3</sup> Our second

<sup>3</sup>Restricting the co-occurrence to a certain window size did not improve the F-Score of *take-all*.

Word-based Features		
Feature	Abbrev.	Illustration/Further Information
bag of words between the mention of <i>healthy</i> and target food item, and the additional words that precede or follow <i>healthy</i> and target food item	word	N/A
Linguistic Features		
Feature	Abbrev.	Illustration/Further Information
Are target food item and <i>healthy</i> within the same clause?	clause	<i>I like chocolate<sub>target</sub>, even though I consider fruits the healthy option for snacks.</i> Feature operates on parse output.
Is there a punctuation mark between target food item and <i>healthy</i> ?	boundary	<i>I know that vegetables are extremely healthy; but I prefer chocolate<sub>target</sub>.</i> Token-level back-off feature to <i>clause</i> .
Is there another food item between target food item and <i>healthy</i> ?	otherFood	<i>We always had healthy meals with lots of vegetables and <u>salad</u>, but this does not mean that we were not allowed to eat chocolate<sub>target</sub>.</i> Token-level back-off feature to <i>clause</i> .
Is target food item in a prominent position?	prom	Prominent positions: e.g. beginning/end of a sentence/subclause.
Is target food item used as a side dish?	side	<i>Broccoli with potatoes<sub>target</sub> is a healthy dish.</i> Patterns from relation type <i>Served-with</i> used in Wiegand et al. (2012a).
Is <i>healthy</i> a predicative adjective relating to target food item?	predRel	<i>Vegetables are healthy.</i>
Is <i>healthy</i> an attributive adjective relating to target food item?	attrRel	<i>I would recommend buying some healthy fat.</i>
Is <i>healthy</i> a predicative adjective?	pred	<i>I really like bananas<sub>target</sub> and they are healthy, too.</i>
Is <i>healthy</i> an attributive adjective?	attr	<i>For that we need to use some kind of fat<sub>target</sub>; I particularly favour the healthy ones.</i>
Does <i>healthy</i> precede target food item?	precede	If <i>healthy</i> precedes the target food item, then this often indicates <i>attributive</i> usage.
Is <i>healthy</i> an attributive adjective of a general food expression (i.e. <i>meal, dish, food</i> , etc.) that is not target food item?	attrFood	<i>Salad is a healthy dish.</i>
Is there some quantification?	quant	<i>100g per day; in moderation; a teaspoon of;</i> a list of 75 quantifying expressions was collected from the web ( <a href="http://rezepte.nit.at/kuechenmasse.html">rezepte.nit.at/kuechenmasse.html</a> ) and <a href="http://de.wikibooks.org/wiki/Kochbuch/_Maßangaben">de.wikibooks.org/wiki/Kochbuch/_Maßangaben</a> ).
Is target food item modified by an attributive adjective other than <i>healthy</i> ?	attrNoH	<i>steamed vegetables; fried potatoes</i>
Is target food item further specified?	spec	<i>bread<sub>target</sub> made of whole grains; cake<sub>target</sub> with low-fat ingredients;</i> Complementary feature to <i>attrNoH</i> (feature detects specifications in the form of contact clauses or prepositional phrases immediately attached to the target food item).
Is there a cue indicating an opinion holder other than the author?	opHolder	<i>Some people claim that chocolate is healthy.</i> This feature relies on a set of predicates indicating the presence of an opinion holder (Wiegand and Klakow, 2011).
Is target sentence a (direct) question?	question	<i>Is chocolate healthy?</i>
Is <i>healthy</i> embedded in some <i>irrealis</i> context?	irrealis	<i>If honey were healthy; I wonder, whether honey is healthy.</i> Translation of the cues used in hedge classification (Morante and Daelemans, 2009).
Is <i>healthy</i> modified by a modal verb?	modal	<i>Honey might be healthy.</i>
Is target food item negated?	negTarget	<i>No cake is healthy.</i> We adapted to German the negation word lists and the scope modeling from Wilson et al. (2005).
Is <i>healthy</i> negated?	negHealth	<i>Chocolate is not healthy.</i> We adapted to German the negation word lists and the scope modeling from Wilson et al. (2005).
Is there any occurrence of a <i>weird</i> word?	weird	<i>Sure, chocolate is veeeeery healthy.</i> Regular expression detecting suspicious reduplications of characters in order to detect irony.
Does the context suggest that <i>healthy</i> is part of a comparison?	comp	We check for typical inflectional word forms (i.e. <i>healthier</i> and <i>healthiest</i> ) and constructions, such as <i>as healthy as</i> .
Does the context of <i>healthy</i> suggest another sense of the word?	sense	Contexts in which <i>healthy</i> has a different meaning (using online dictionaries, such as <a href="http://www.duden.de/rechtschreibung/gesund">www.duden.de/rechtschreibung/gesund</a> and <a href="http://de.wiktionary.org/wiki/gesund">de.wiktionary.org/wiki/gesund</a> ).
Number of positive/negative polar expressions (excluding mentions of <i>healthy</i> )	polar*	Usage of the German <i>PolArt</i> sentiment lexicon (Klenner et al., 2009).
Number of near synonyms of (un)healthy	syno*	Examples for healthy: <i>high in vitamin, tonic</i> , etc.; examples for unhealthy: <i>carcinogenic, harmful</i> , etc. (manually compiled list of 99 synonyms by an annotator <b>not</b> involved in feature engineering).
Number of diseases	disease*	411 entries, created with the help of the web ( <a href="http://bildung.wikia.com/wiki/Alphabetische_Liste_der_Krankheiten">bildung.wikia.com/wiki/Alphabetische_Liste_der_Krankheiten</a> ).
Task-specific Knowledge-based Features using a Healthiness Lexicon		
Feature	Abbrev.	Illustration/Further Information
Is target food item <i>a priori</i> healthy?	prior*	Feature employs the healthiness lexicon from Wiegand et al. (2012b).
Is target food item <i>a priori</i> unhealthy?		
Number of food items (excluding target food item) that are <i>a priori</i> healthy	priorCont*	Feature employs the healthiness lexicon from Wiegand et al. (2012b).
Number of food items (excluding target food item) that are <i>a priori</i> unhealthy		

\*: there exist two features which differ in the context they consider: (a) only target sentence (indicated by suffix *-TS*) (b) entire context (indicated by suffix *-EC*)

Table 2: Description of the feature set; the set contains several cue word lists, in order to **avoid overfitting**, we either translated existing resources from English or used diverse web-resources that are **not** related to our dataset.

Features	HLTH			UNHLTH		
	Pre	Rec	F1	Pre	Rec	F1
take-all ( <i>baseline 1</i> )	20.3	<b>100.0</b>	33.7	6.9	<b>100.0</b>	13.0
prior ( <i>baseline 2</i> )	28.0	87.3	42.3	29.7	44.0	35.3
priorCont	21.2	96.9	34.7	14.3	34.8	20.3
prior+priorCont	28.0	86.9	42.3	29.7	44.0	35.3
word	35.9	66.5	46.6	39.7	42.5	41.0
linguistic	38.3	66.1	48.3	35.9	43.5	39.1
word+linguistic	40.2	63.6	49.1*	40.9	47.1	43.4*
word+prior	38.1	70.1	49.2°	46.7	43.3	44.7
word+priorCont	35.0	65.3	45.5	40.0	42.9	41.0
word+prior+priorCont	37.4	70.8	48.8°	<b>46.8</b>	42.8	44.4
word+linguistic+priorCont	41.4	64.3	50.2	42.8	42.1	41.7
word+linguistic+prior	44.1	68.3	53.3°†‡	44.8	60.5	<b>51.1</b> °†‡
all features	44.5	69.3	<b>53.9</b> °†‡	42.9	63.5	51.0°†‡
rule-based	<b>53.4</b>	17.9	26.8	45.0	11.0	17.7

significantly better than *word*\* at  $p < 0.1$ ° at  $p < 0.05$ ; better than *word+linguistic*† at  $p < 0.05$ ; better than *word+prior*‡ at  $p < 0.05$  (paired t-test)

Table 4: Comparison of different feature sets.

baseline is *prior* (see §5.2 for motivation).

*take-all* has optimal recall but a very poor precision. The second baseline *prior* is notably better. *prior* may help to distinguish between *HLTH* and *UNHLTH* but it does not contribute to distinguishing these classes from the rest of the relation types (Table 1).

If we turn to the features that largely exploit contextual information, i.e. *word* and *linguistic* (§5.1), we find that both features are better than the previous features. This is an indication that learning from text is effective. The same can be said about *word+linguistic* and *word+prior*, which also outperform *word*. *word+linguistic+prior* is the best feature set outperforming both *word+linguistic* and *word+prior*. We conclude that all of the three groups of features we presented in §5 are relevant for this task.

In terms of recall and F-score the supervised classifier always outperforms the rule-based classifier. This does not come as a surprise as the supervised classifier learns from labeled training data while the rule-based classifier is unsupervised. On the other hand, we also find that the precision of the rule-based classifier largely outperforms our best supervised classifier on *HLTH*.

The fact that the best overall F-score achieved is not higher may be ascribed to the heavy noise (spelling/grammar mistakes) contained in our web-data. However, we believe that even with those data we can show the relative effectiveness of the different feature types which is the most relevant aspect in our *proof-of-concept* investigation.

Class	Features
HLTH	prom, attrNoH, predRel, comp, negHealth, <i>negative polarEC</i> , sense, opHolder, irrealis
UNHLTH	negHealth, negTarget, attrRel, comp, diseaseTS, <i>negative polarEC</i>

Table 5: List of the best subset of linguistic features (Table 2) for each individual class.

### 7.1.2 Inspection of Linguistic Features

Table 5 shows the best performing feature subset using a best-first forward selection as implemented in *Weka* (Witten and Frank, 2005). The table shows that diverse features are important including features to detect restricted relations (§4.3.1) (i.e. *attrNoH*) or comparisons (i.e. *comp*), features to distinguish predicative from attributive adjectives for the detection of unspecified intersection (§4.3.2) (i.e. *predRel* and *attrRel*), various features to determine contextual embedding (i.e. *opHolder*, *irrealis* and *negHealth*) and sentiment information (i.e. *negative polarEC*).

### 7.1.3 Detecting Anti-Prior Healthiness

We now take a closer look at *anti-prior* instances which are utterances in which the relation expressed is opposite to the relation that one would *a priori* assume, e.g. *chocolate is healthy* instead of *chocolate is unhealthy*. In our gold standard, we identified these instances with the help of the actual (manually assigned) label and our healthiness lexicon (§5.2).<sup>4</sup> Such instances may be very interesting to extract, even though they are rare (15% on *HLTH* and *UNHTLH*). Previously, supervised classifiers with *word+prior* produced similar performance as classifiers with *word+linguistic* (Table 4). Since linguistic features are fairly expensive to produce, the prior knowledge of healthiness seems an attractive alternative. But this is misleading. Table 6 displays the recall (by supervised classification) on only anti-prior instances and shows that the usage of *prior* which, in isolation, would detect none of these instances, gives a much lower recall than *linguistic* when added to *word*. Therefore, *word+linguistic* would be the preferable feature set if one had to choose between *word+prior* and *word+linguistic*.

<sup>4</sup>Whenever HLTH co-occurs with prior unhealthiness (according to the healthiness lexicon) or UNHLTH co-occurs with prior healthiness, there is an anti-prior instance.

Feature Set	word+prior	word+linguistic
Recall	17.2	54.6

Table 6: Recall on *anti-prior* instances.

## 7.2 Aggregate Classification

Finally, we automatically rank food items according to healthiness based on the aggregate of text mentions. Ideally, the ranking should separate healthy from unhealthy food items. We want to know whether with our text corpus and contextual classification, one can actually approximate a correct prior healthiness. Aggregate classification means that we make a healthiness prediction for a specific food item based on *all* text mentions of that food item co-occurring with the word *healthy*. It may be easier to achieve a robust aggregate classification than a robust individual classification. This is because in aggregate-based tasks, there is a certain degree of redundancy contained in the data, as instances of a group of utterances (belonging to the same food item) may often comprise similar information. For such classifiers, one should focus on a higher precision since a reasonable recall is enabled by the redundancy in the data.

Our baseline **RAW** is completely unsupervised and does not include any linguistic processing. We use the *Pointwise Mutual Information (PMI)* which is estimated on our large web corpus (§3).<sup>5</sup>

$$PMI(\text{food item}, \text{healthy}) = \log \frac{P(\text{food item}, \text{healthy})}{P(\text{food item})P(\text{healthy})} \quad (1)$$

For the automatic classification, we consider **LEARN** which uses the output of the supervised classifier comprising the features *word+linguistic* (we *must* exclude the feature *prior* as this would include the knowledge we want to predict automatically in this experiment)<sup>6</sup> while **RB** is the output of the rule-based classifier we presented in §6 (which does not contain *prior* as a feature either).

In order to convert the classifications of individual utterances for a target food item (by **LEARN** and **RB**) to one ranking score (according to which we rank all the target food items), we simply compute the ratio between instances predicted to be healthy and those predicted to be unhealthy:

$$\text{score}_{LEARN/RB}(\text{food item}) = \frac{\#HLTH_{predicted}(\text{food item})}{\#UNHLTH_{predicted}(\text{food item})} \quad (2)$$

<sup>5</sup>For  $P(\text{food item}, \text{healthy})$ , we consider all *sentences* in which the target food item and *healthy* co-occur.

<sup>6</sup>We train for each target food item a classifier using only the instances with the other target food items as training data.

<b>RAW</b>	wholemeal product > fat > colza oil > vegetables > tea > protein > olive oil > honey > meat > sugar > salad > bread > chocolate > potato > rice > banana > cake > water > egg
<b>LEARN</b>	banana > olive oil > wholemeal product > tea > colza oil > salad > vegetables > protein > potato > chocolate > meat > bread > rice > water > sugar > cake > egg > fat > honey
<b>RB</b>	potato > protein > wholemeal product > banana > olive oil > vegetables > bread > salad > water > tea > colza oil > rice > honey > egg > chocolate > fat > meat > sugar > cake

Table 7: Aggregate ranking; **green** denotes (actual) healthy items, **red** (actual) unhealthy items.

where  $\#HLTH_{predicted}(\text{food item})$  are the number of instances the classifier predicts the label *HLTH* for the target food item while  $\#UNHLTH_{predicted}(\text{food item})$  are the number of instances labeled as *UNHLTH*, respectively.

Table 7 shows the results of the three rankings. The actual labels are derived from the healthiness lexicon (§5.2). The table clearly shows that the ranking produced by **RAW** contains most errors. *fat* is the second most highly ranked food item. This can be explained by the high proportion of *INTERS* (§4.3.2) among the co-occurrences of *fat* and *healthy* (almost 50%). **LEARN** and **RB** produce a better ranking, thus proving that a contextual (linguistic) analysis is helpful for this task. **RB** also outperforms **LEARN** presumably because of its much higher precision (as measured for individual classification in Table 4: 53.4% vs. 40.2% for *HLTH* and 45.0% vs. 40.9% for *UNHLTH*).

## 8 Conclusion

We presented a first step towards contextual healthiness classification of food items. For this task, we introduced a new annotation scheme. Our annotation revealed that many different linguistic phenomena are involved. Thus, this problem can be considered an interesting task for NLP. We demonstrated that a linguistic analysis is not only necessary for classifying individual utterances but also for ranking food items based on an aggregate of text mentions.

## Acknowledgements

This work was performed in the context of the Software-Cluster project EMERGENT. Michael Wiegand was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. “01IC10S01”. The authors would like to thank Stephanie Köser and Eva Lasarczyk for annotating the dataset presented in this paper. We would also like to thank Benjamin Roth for interesting discussions.

## References

- Victor Chahuneau, Kevin Gimpel, Bryan R. Routledge, Lily Scherlis, and Noah A. Smith. 2012. Word Salad: Relating Food Prices and Descriptions. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, pages 1357–1367, Jeju Island, Korea.
- Aaron M. Cohen and William R. Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6:57 – 71.
- Ernesto Diaz-Aviles, Avar Stewart, Edward Velasco, Kerstin Denecke, and Wolfgang Nejdl. 2012. Epidemic Intelligence for the Crowd, by the Crowd. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, Dublin, Ireland.
- Marco Fisichella, Avar Stewart, Alfredo Cuzzocrea, and Kerstin Denecke. 2011. Detecting Health Events on the Social Web to Enable Epidemic Intelligence. In *Proceedings of the International Symposium on String Processing and Information Retrieval (SPIRE)*, pages 87–103, Pisa, Italy.
- Thorsten Joachims. 1999. Making Large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press.
- Manfred Klenner, Stefanos Petrakis, and Angela Fahrni. 2009. Robust Compositional Polarity Classification. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 180–184, Borovets, Bulgaria.
- J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.
- 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 Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL/IJCNLP)*, pages 1003–1011, Singapore.
- Roser Morante and Walter Daelemans. 2009. Learning the Scope of Hedge Cues in Biomedical Texts. In *Proceedings of the BioNLP Workshop*, pages 28–36, Boulder, CO, USA.
- Robert Munro, Lucky Gunasekara, Stephanie Nevins, Lalith Polepeddi, and Evan Rosen. 2012. Tracking Epidemics with Natural Language Processing and Crowdsourcing. In *Proceedings of the Spring Symposium for Association for the Advancement of Artificial Intelligence (AAAI)*, pages 52–58, Toronto, Canada.
- Fred Popowich. 2005. Using Text Mining and Natural Language Processing for Health Care Claims Processing. *SIGKDD Explorations*, 7(1):59–66.
- Anna Rafferty and Christopher D. Manning. 2008. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the ACL Workshop on Parsing German (PaGe)*, pages 40–46, Columbus, OH, USA.
- Marina Sokolova and Victoria Bobicev. 2011. Sentiments and Opinions in Health-related Web Messages. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 132–139, Hissar, Bulgaria.
- Manabu Torii, Lanlan Yin, Thang Nguyen, Chand T. Mazumdar, Hongfang Liu, David M. Hartley, and Noele P. Nelson. 2011. An exploratory study of a text classification framework for internet-based surveillance of emerging epidemics. *International Journal of Medical Informatics*, 80(1):56–66.
- Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. ICWSM - A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, Washington, DC, USA.
- Willem Robert van Hage, Sophia Katrenko, and Guus Schreiber. 2005. A Method to Combine Linguistic Ontology-Mapping Techniques. In *Proceedings of International Semantic Web Conference (ISWC)*, pages 732 – 744, Galway, Ireland. Springer.
- Willem Robert van Hage, Hap Kolb, and Guus Schreiber. 2006. A Method for Learning Part-Whole Relations. In *Proceedings of International Semantic Web Conference (ISWC)*, pages 723 – 735, Athens, GA, USA. Springer.
- Willem Robert van Hage, Margherita Sini, Lori Finch, Hap Kolb, and Guus Schreiber. 2010. The OAEI food task: an analysis of a thesaurus alignment task. *Applied Ontology*, 5(1):1 – 28.
- Michael Wiegand and Dietrich Klakow. 2011. The Role of Predicates in Opinion Holder Extraction. In *Proceedings of the RANLP Workshop on Information Extraction and Knowledge Acquisition (IEKA)*, pages 13–20, Hissar, Bulgaria.
- Michael Wiegand, Benjamin Roth, and Dietrich Klakow. 2012a. Web-based Relation Extraction for the Food Domain. In *Proceedings of the International Conference on Applications of Natural Language Processing to Information Systems (NLDB)*, pages 222–227, Groningen, the Netherlands. Springer.
- Michael Wiegand, Benjamin Roth, Eva Lasarczyk, Stephanie Köser, and Dietrich Klakow. 2012b. A Gold Standard for Relation Extraction in the Food Domain. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 507–514, Istanbul, Turkey.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 347–354, Vancouver, BC, Canada.
- Ian Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, San Francisco, US.
- GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 427–434, Ann Arbor, MI, USA.



# Learning a Replacement Model for Query Segmentation With Consistency in Search Logs

Wei Zhang<sup>§\*</sup>, Yunbo Cao<sup>‡</sup>, Chin-Yew Lin<sup>‡</sup>, Jian Su<sup>§</sup>, Chew-Lim Tan<sup>†</sup>

<sup>§</sup>Institute for Infocomm Research, <sup>‡</sup>Microsoft Research Asia

<sup>†</sup>National University of Singapore

{zhangw3, sujian}@i2r.a-star.edu.sg

{yunbo.cao, cyl}@microsoft.com, tancl@comp.nus.edu.sg

## Abstract

Query segmentation is to split a query into a sequence of non-overlapping segments that completely cover all tokens in the query. The majority of methods are unsupervised, however, they are usually not as accurate as supervised methods due to the lack of guidance from labeled data. In this paper, we propose a new paradigm of *learning a replacement model with consistency* (LRMC), to enable unsupervised training with guidance from search log data. In LRMC, we first assume the existence of a base segmenter (an implementation of any existing approach). Then, we utilize a key observation that queries with a similar intent tend to have *consistent* segmentations, to automatically collect a set of labeled data from the outputs of the base segmenter by leveraging search log data. Finally, we employ the auto-collected data to train a replacement model for selecting the correct segmentation of a new query from the outputs of the base segmenter. The results show LRMC can improve state-of-the-art methods by an F-Score of around 7%.

## 1 Introduction

Nowadays *keyword queries* have been adopted as the de-facto query interface by most search engines. Query tokens are not independent or unordered symbols but rather ordered and structured words and phrases with syntactic relationships. Understanding the structure of a query is crucial for achieving better search performance. Such an understanding will also ease other search related applications such as query suggestion and rewriting, where one is able to work on semantic concepts instead of individual tokens. *Query segmentation* (QS), a process of splitting a query into a sequence of non-overlapping segments that

completely cover all tokens, aims to address these challenges. It requires that every segment rendered is a phrase or a semantic unit. For example, given a query “download adobe writer”, four different ways of segmentation are possible. The challenge is to determine which one is correct.

The majority of QS methods are unsupervised, however, they are not as accurate as supervised methods due to lack of guidance from labeled data. On the other hand, supervised models suffer from the problems: (1) new phrases/words are introduced on the web daily, which quickly invalidate static supervised models trained on a certain manually labeled set; (2) it is not feasible to develop a set of labeled data covering all domains on the web. In this paper, we propose a paradigm of *learning a replacement model with consistency* (LRMC), to enable unsupervised training and it improves various unsupervised QS systems.

LRMC first assumes the existence of a base segmentation system (hereafter referred to as ‘*base segmenter*’) which can output top- $n$  segmentations for any query. Then it tries to learn a *replacement model* capable of selecting the correct segmentation of a new query (if one exists) from the output of the base segmenter. Our study on three state-of-the-art systems (Section 5.2) shows that for more than 35% of queries the correct segmentations are not ranked as top-1 but included in the top-5 results of the base segmenter, which implies the potential of LRMC. The keys to our proposal include: (a) how to *automatically* acquire labeled data (i.e., for a query in the labeled data, what its correct segmentation is) and then (b) how to use the labeled data to learn the replacement model.

Our method for the automatic acquisition of the labeled data is motivated by the observation: *Queries with a similar intent tend to have consistent segmentation results*. In this paper, we say that a set of queries have similar intents if and only if they lead to the same set of web documents (i.e., clicks). For example, when issuing to a web

\*Wei Zhang did this work when he was an intern at Microsoft Research Asia.

queries	Rank-1 Segmentation Result	Rank-2 Segmentation Result
download adobe writer free adobe writer download free adobe writer	download adobe   writer <b>free   adobe writer   download</b> <b>free   adobe writer</b>	<b>download   adobe writer</b> free   adobe   writer   download free adobe   writer

Table 1: Segmentation results for queries with a similar intent (Results in bold are considered correct.)

search engine any of the three queries in Table 1, we search for the same set of web pages which can provide ‘free download of Adobe writer’. We denote such a set of queries as ‘*query intent set*’. For the queries in the same *query intent set*, naturally we wish to explain them in the same way and thus require that their segmentations be consistent with each other. We say that  $q_1$  and  $q_2$  are inconsistent in segmentation if there exist more than one common subsequence of tokens having different segment boundaries. In Table 1, we also include the top-2 segmentation results that can possibly be generated by any base segmenter. If we check only the ‘rank-1’ results, we observe that the segmentation ‘download adobe | writer’ disagrees with the other two. This means that we interpret the same sequence of tokens differently for different queries with a same intent, which is not what we expect to have. Instead, we expect to have the *bolded* segmentations in which none of the individual segments for one query disagrees with the segments for another query. In this paper, we propose two methods for selecting such correct segmentations from top- $n$  segmentation results that are about the same *query intent sets*. With these methods, we can automatically build up a training data set, which allows us to train a reliable model.

The replacement model concerns about *whether or not a ‘rank-1’ segmentation  $S^a$  generated by a base segmenter should be replaced by a ‘rank- $k$ ’ ( $k > 1$ ) segmentation  $S^b$* . The decision of the replacement can be made by collectively considering one or multiple local transformations in the form of ‘ $w_i w_{i+1} \mapsto w_i | w_{i+1}$ ’ or ‘ $w_i | w_{i+1} \mapsto w_i w_{i+1}$ ’. ‘ $w_i w_{i+1} \mapsto w_i | w_{i+1}$ ’ means that  $S^a$  does not include a segment boundary between tokens  $w_i$  and  $w_{i+1}$  and  $S^b$  does; Similarly, ‘ $w_i | w_{i+1} \mapsto w_i w_{i+1}$ ’ means the reverse. For example, for the first query in Table 1, we can have the local transformations ‘download adobe  $\mapsto$  download | adobe’ and ‘adobe | writer  $\mapsto$  adobe writer’. The proposed model estimates the score of every local transformation using a binary classifier and then aggregates the individual scores to reach its final decision.

We conduct extensive experiments using two public data sets. The results show that (a) our

method for automatically constructing a set of labeled data with a base segmenter and a set of query intent sets as inputs is effective, capable of discovering correct segmentations missed by the evaluated base segmenters for more than 20% of queries (See **M2** in terms of  $Acc^{qry}$  in Table 3); and (b) our replacement model benefits existing QS approaches and boosts their performance significantly (e.g. the improvement of  $> 7\%$  F-Score on the data WQ10-Majority in Table 4).

We summarize our contributions as follows: (1) on the basis of the observation that queries with a similar intent tend to have consistent segmentations, we propose a method for automatically collecting from search log data a set of labeled data for QS. The method first groups queries in search log data into what we call a ‘query intent set’ and then select correct segmentations by examining the consistency among segmentations for the queries in the same ‘query intent sets’. (2) With the automatically-collected data, we develop a ‘replacement model’ for the purpose of checking whether or not a ‘rank-1’ segmentation generated by a base segmenter should be replaced by a ‘rank- $k$ ’ ( $k > 1$ ) segmentation. (3) We conduct extensive experiments with two publicly available data sets and show that our proposal can effectively boost the performance of state-of-the-art systems (Hagen et al., 2010; Hagen et al., 2011).

## 2 Related Work

Bergsma and Wang (2007) considered the decision to segment or not between each pair of adjacent words as a binary classification problem. Guo et al. (2008), Yu and Shi (2009), and Kiseleva et al. (2010) used methods based on CRF. As the cost of obtaining labeled data is high, they are usually not feasible to develop a set of labeled data covering all the domains on the web and then train a scalable QS model for web search.

The work for web-scale QS are usually unsupervised and utilized various statistics such as mutual information (MI) and frequency count collected from various sources such as web data, query logs, and etc (Risvik et al., 2003; Jones et al., 2006; Huang et al., 2010; Zhang et al., 2009). Li et al. (2011) also used the language model estimated

from click-through documents to backoff the generating process of QS. Tan and Peng (2008) used n-gram frequencies from a large web corpus as well as Wikipedia. Hagen et al. (2010) showed that the raw n-gram could be exploited with an appropriate normalization scheme and achieved surprisingly good accuracy. Later, they enriched the work by including the use of Wikipedia (Hagen et al., 2011). In our evaluation, we compare our proposal with the last two work which represent state-of-the-art.

Our proposal is orthogonal to all the above approaches. LRMC assumes the existence of a base segmenter (an implementation of any above approaches) and it focuses on how to leverage search log data to learn a replacement model for improving the output of base segmenters.

### 3 Problem Settings

**QS.** Let  $q = [w_1, w_2, \dots, w_n]$  denote a query consisting of  $n$  keywords. A segment  $s = [w_i, \dots, w_j] (1 \leq i \leq j \leq n)$  is a subsequence of the query. A segmentation  $S = [s_1|s_2|\dots|s_K]$  for query  $q$  is then defined as a sequence of non-overlapping segments. ‘|’ denotes a segmentation boundary. If we assume there is no order dependency of  $s$ , we can then treat  $S$  as a set  $\{s_k\}_{k=1}^K$ .

**Query Intent.** There exist many definitions on query intent. In this paper we introduce an operational definition on query intent.

**Definition 1** *The query intent( $s$ ) of a query  $q$  is defined as the set of URLs ( $Urls(q)$ ) which are clicked for  $q$  by users of a web search engine.*

Because most queries are ambiguous or multi-faceted (Clarke et al., 2009), we manage to restrict the number of intents into one or a few by grouping more queries together, which leads to the definition of ‘query intent set’.

**Definition 2** *A query intent set  $Q^{INT}$  is a set of queries satisfying the following conditions:*

- a)  $\bigcap_{q \in Q^{INT}} Urls(q) \neq \emptyset$ ;
- b)  $|Q^{INT}| > c$ .

where  $|Q^{INT}|$  denotes the number of elements in  $Q^{INT}$ , and  $c$  is a parameter to control how specific a query intent is; a larger value for  $c$  usually means that the query intent is more specific and thus less ambiguous.

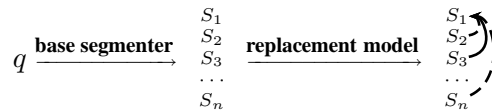
Query intent sets used in our experiments will be detailed in Section 5.1.

## 4 Our Proposal

### 4.1 Overview of the Proposed Paradigm

First, the paradigm LRMC assumes the existence of a base segmenter that is able to output top- $n$  segmentations for any query. Then it tries to learn a *replacement model* capable of replacing the rank-1 segmentation generated by this base segmenter with one rank- $k$  ( $k > 1$ ) segmentation.

LRMC can be illustrated by the following flowchart. First, a query  $q$  is fed into a base segmenter. As a result, a set of segmentations  $\{S_i\}_{i=1}^n$  regarding  $q$  are generated. Subscript  $i$  denotes the rank of the corresponding segmentation. Next,  $\{S_i\}_{i=1}^n$  are fed into a replacement model. The replacement model tries every possible replacement  $S_i$  ( $i > 1$ ) for the rank-1 segmentation  $S_1$  (as indicated by the curved arrows). The trial ends with two possible results: (a) None of the replacements is valid ( $S_1$  cannot be replaced); and (b) one segmentation  $S_i^*$  ( $i^* > 1$ ) is the most likely replacement and thus chosen as the final segmentation for  $q$  (e.g., the replacement of the solid curve).



LRMC is motivated by the following observation: for most cases, the correct segmentation for a query is included in its top- $n$  segmentation results already. Usually, there are not that many likely segmentations for a query and thus correct segmentations cannot be ranked too low by a base segmenter. For example, for any base segmenter in our experiment, more than 93% of queries can have a segmentation that is agreed upon by at least one of the annotators in its top-5 results. Given this observation, what we have to do is not to generate or propose a new segmentation, but to tell which segmentation is correct in the top- $n$  results.

Next, we detail how the replacement model is learned. Specifically, we first introduce how we automatically extract from search log data a set of labeled data with ‘consistency’ as a guidance and then explain how a ‘replacement model’ can be learned from this data set.

### 4.2 Consistency as Supervision

Assume that we have a *query intent set*  $Q^{INT} = \{q_i\}_{i=1}^m$ . With a base segmenter, we generate the top- $n$  segmentation results  $\{S_{ij}\} (1 \leq j \leq n)$  for

each query  $q_i$ , which forms the following matrix:

$$S_{Q^{INT}} = \begin{pmatrix} S_{11} & \underline{S_{12}} & S_{13} & \cdots & S_{1n} \\ S_{21} & \underline{S_{22}} & S_{23} & \cdots & S_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \underline{S_{m1}} & S_{m2} & S_{m3} & \cdots & S_{mn} \end{pmatrix} \begin{array}{l} \leftarrow q_1 \\ \leftarrow q_2 \\ \cdots \\ \leftarrow q_m \end{array}$$

What we manage to achieve is to collect a set of ‘labeled’ data from  $\{S_{ij}\}$ . In the ‘labeled’ data, each query  $q_i$  has only one segmentation  $S_{ij^*}$  ( $S_{ij^*} \in \{S_{ij}\}_{j=1}^n$ ), which we consider ‘correct’. We make use of two types of strategies to choose the ‘correct’ segmentations from  $\mathbf{S}_{Q^{INT}}$  (e.g., those underlined ones in  $\mathbf{S}_{Q^{INT}}$ , namely  $S_{12}, S_{22}, \dots, S_{m1}$ ).

Before explaining the two strategies, let us first introduce how we measure the consistency between two segmentations. The *consistency*  $cst(S, S')$  between segmentations  $S$  and  $S'$  is defined as the number of segments they share, i.e.,

$$cst(S, S') = |S \cap S'| \quad (1)$$

The first strategy (**M1**) that we use as ‘supervision’ for the acquisition of the labeled data is as follows: The correct segmentations for the  $m$  queries in the same *query intent set* should be very consistent with or similar to each other although the segmentations cannot be exactly the same. Thus, the correct segmentations can be chosen with the objective function:

$$(j_1^*, \dots, j_m^*) = \arg \max_{1 \leq j_1, \dots, j_m \leq n} \sum_{1 \leq i < i' \leq m} cst(S_{ij_i}, S_{i'j_{i'}}) \quad (2)$$

where  $j_i^*$  denote the index (or rank) of the correct segmentation for query  $q_i$ .

The other strategy (**M2**) is on the basis of the observation: Although at most only one top- $n$  segmentation can be correct for a query, most segmentations are not totally incorrect, i.e., they include some correct segments while having some incorrect segments as well. Thus, those incorrect segmentations also provide some clues about what can be correct. In addition, as the choices for ‘incorrect segment’ are usually more than those for correct segment, it is relatively hard for incorrect segments to converge to a few. As a result, a correct segment should be more popular than any one single incorrect segment. Given this discussion, we can have the second objective function:

$$(J_1^*, \dots, J_m^*) = \arg \max_{1 \leq j_1, \dots, j_m \leq n} \left( \sum_{1 \leq i, i' \leq m} \sum_{1 \leq j' \leq n} cst(S_{ij_i}, S_{i'j'}) \right) - \sum_{1 \leq i \leq m} cst(S_{ij_i}, S_{ij_i}) \quad (3)$$

Note that  $cst(S_{ij_i}, S_{ij_i}) = |S_{ij_i}|$ . Given one selected segmentation  $S_{ij_i}$ , the objective is to sum up the consistencies between itself and any of the rest in matrix  $S_{Q^{INT}}$ . Thus, by this objective, we choose the segmentations whose segments are agreed with by most top- $n$  segmentations.

Both strategies assume that correct segments are more popular than incorrect segments in the top- $n$  output of one reasonably-performing base segmenter. Both strategies will fail if the assumption is not true. Our experiments in Section 5.2, in which both strategies are able to find more correct segmentations than the base segmenters, can be seen as a support for the assumption.

### 4.3 Replacement Model

The replacement model is to tell whether or not a segmentation ranked as top-1 by a base segmenter should be replaced by another segmentation with a rank of  $j$  ( $1 < j \leq n$ ). For example, we have a query  $q$  whose top- $n$  segmentations are  $\{S_{qj}\}_{j=1}^n$ . Then, the input of the replacement model will be a possible replacement  $S_{q1} \mapsto S_{qj}$  ( $j > 1$ ) and the output will be a label ‘1’ or ‘0’. Label ‘1’ means  $S_{q1}$  should be replaced by  $S_{qj}$  and ‘0’ means ‘not’.

With that in mind, we can then make use of ‘consistency’ to create a labeled data set. For example, if query  $q$  belongs to a query intent set  $Q^{INT}$  and its correct segmentation chosen by the objective (2) or (3) is  $S_{qj^*}$ , we can generate the labeled instance(s) as follows:

$$D_q = \begin{cases} \{(S_{q1} \mapsto S_{qj}, 0)\}_{j \neq 1} & \text{if } j^* = 1 \\ \{(S_{q1} \mapsto S_{qj^*}, 1)\} & \text{otherwise} \end{cases} \quad (4)$$

By combining all such data sets together, we then have the final labeled data set  $D = \bigcup_q D_q$ . Note that query  $q$  can come from multiple query intent sets (not just one single set).

Next, we explain how to use the above training data to learn a replacement model.

The decision of whether or not to do the replacement of  $S_{q1} \mapsto S_{qj}$  can be made by collectively considering one or multiple local transformations in the form of ‘ $w_i w_{i+1} \mapsto w_i | w_{i+1}$ ’ or ‘ $w_i | w_{i+1} \mapsto w_i w_{i+1}$ ’. ‘ $w_i w_{i+1} \mapsto w_i | w_{i+1}$ ’ means that  $S_{q1}$  does not include a segment boundary between tokens  $w_i$  and  $w_{i+1}$  and  $S_{qj}$  does; ‘ $w_i | w_{i+1} \mapsto w_i w_{i+1}$ ’ means the reverse.

Let  $T(S_{q1} \mapsto S_{qj})$  denote the set of all possible local transformations from  $S_{q1}$  to  $S_{qj}$  and  $\mathbf{x}$  denote

one element from the set (i.e., one local transformation). If we know the likelihood  $f(\mathbf{x})$  of every individual transformation  $\mathbf{x}$  being valid, the score of replacing  $S_{q_1}$  by  $S_{q_j}$  can then be estimated as  $\sum_{\mathbf{x} \in T(S_{q_1} \mapsto S_{q_j})} f(\mathbf{x})$ .

The likelihood of a local transformation  $\mathbf{x}$  being valid can be estimated with a binary classifier. We employ SVM as the classifier. Given an instance  $\mathbf{x}$ , SVM assigns a score to it based on  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ , where  $\mathbf{w}$  denotes a weight vector and  $b$  denotes an intercept. Given a replacement  $(S_{q_1} \mapsto S_{q_j}, y)$  where  $y \in \{0, 1\}$ , a set of labeled data for the binary classifier is prepared as:  $\{\mathbf{x}, y\}_{\mathbf{x} \in T(S_{q_1} \mapsto S_{q_j})}$ . By considering all the replacements in  $D$ , we will have a final training data set  $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$  for SVM.

On the basis of that, we can do the replacement as follows: If for certain  $j$  ( $j > 1$ )  $\sum_{\mathbf{x} \in T(S_{q_1} \mapsto S_{q_j})} f(\mathbf{x}) > 0$ , we will use the segmentation with  $\arg \max_{1 < j \leq n} \sum_{\mathbf{x} \in T(S_{q_1} \mapsto S_{q_j})} f(\mathbf{x})$  as its index to replace the top-1 segmentation; Otherwise, we will keep using the top-1 segmentation.

#### 4.4 Learning Features

In this section, we describe the features for representing a local transformation  $\mathbf{x}$ , which is in the form of either  $'w_{i_0} w_{i_0+1} \mapsto w_{i_0} | w_{i_0+1}'$  or  $'w_{i_0} | w_{i_0+1} \mapsto w_{i_0} w_{i_0+1}'$ . We utilize four categories of features which are possible indicators of a transformation, representing a variety of information such as lexical, syntactic, semantic and etc.

**Contextual Features: Lexical.** The left and right tokens around the decision position,  $w_{i_0}$  and  $w_{i_0+1}$ , are a good signal of the transformation. In the example of “google desktop | download”, the token ‘download’ is separated from its left neighbor. Such common query tokens in the training data with the property of usually being separated from or being connected to its left/right neighbor can help predict new transformations (e.g. “adobe writer download  $\mapsto$  adobe writer | download”). On the basis of this observation, we adopt the left token  $w_{i_0}$  and the right token  $w_{i_0+1}$  as the features for representing a transformation  $\mathbf{x}$ . Furthermore, sometimes one word alone can not perfectly characterize a transformation. For example, to reject the transformation “diet plan  $\mapsto$  diet | plan”, we have to use the token bigram  $\langle \text{diet plan} \rangle$ . Thus, we include all the token bigrams in the form of  $\langle w_{i_0} w_{i_0+1} \rangle$  as features as well. As we all know, lexical features usually suffer from a data sparse-

ness issue when used in various tasks (Bagga and Baldwin, 1998; Sriram et al., 2010). Fortunately, the web-scale training data we collect from *query intent sets* (Section 4.2) enables us to have a good coverage of lexical features.

**Contextual Features: POS Tag.** Bergsma et al. (2007) show that part-of-speech (POS) tags are useful in their segmentation classification. We also exploit the POS tag pair of  $w_{i_0}$  and  $w_{i_0+1}$  as features. For example, intuitively, “NN NN  $\mapsto$  NN | NN” is more likely to occur than “JJ NN  $\mapsto$  JJ | NN”. The POS tags that we consider include all types of POS tags. Note that this is different from Bergsma et al. (2007). As their segmentation model only takes care of noun phrase queries, their POS tags are restricted to determiners, adjectives, and nouns. The POS tagger by (Roth and Zelenko, 1998) is used in this paper.

**Mutual Information (MI).** Following previous work (Section 2), we also adopt MI between  $w_{i_0}$  and  $w_{i_0+1}$  as our feature. The work (Bergsma and Wang, 2007) also considered the case of a noun phrase with multiple modifiers (e.g. “female bus driver”). To make the segmentation decision between ‘female’ and ‘bus’,  $MI(\text{‘female’}, \text{‘driver’})$  is more suitable to represent the information of not separating them than  $MI(\text{‘female’}, \text{‘bus’})$ . Thus, we also incorporate  $MI(w_{i_0-1}, w_{i_0+1})$  and  $MI(w_{i_0}, w_{i_0+2})$  into our feature set.

Most previous work on QS only can use word-based MI as introduced above. However, in some cases, the MI between tokens can not provide sufficient information for a segmentation decision. For instance, assume that we have the following two queries with their correct segmentations: (1) “download | call of duty | free”; (2) “duty free | shops | sfo”. Only using the token-based mutual information  $MI(\text{‘duty’}, \text{‘free’})$  can not discriminate the two queries from each other and thus can not give different segmentations for ‘duty free’ in the two queries. In our work, as the query has been segmented by a base segmenter, we propose to also use the segment-based MI. In the ‘duty free’ example,  $MI(\text{‘call of duty’}, \text{‘free’})$  will be incorporated for the transformation decision related to “download | call of duty free  $\mapsto$  download | call of duty | free”, where the token-based  $MI(\text{‘duty’}, \text{‘free’})$  does not work.

**Semantic Features.** We define the semantic features on the basis of segments. For a transformation  $'w_{i_0} w_{i_0+1} \mapsto w_{i_0} | w_{i_0+1}'$ , let us denote the

segment including  $w_{i_0}w_{i_0+1}$  (in the first segmentation) as  $s_1$  and denote the segments including  $w_{i_0}$  and  $w_{i_0+1}$  separately (in the second segmentation) as  $s_2$  and  $s_3$ , respectively. To obtain semantic labels for the above three segments, we make use of a web-scale knowledge base of entities, namely Freebase (Bollacker et al., 2008). First, we map the three segments to the Freebase entities by string matching, and then use the names or aliases of the associated categories of the mapped entities as their semantic labels. Finally, the semantic labels for  $s_1$ ,  $s_2$  and  $s_3$  are used as features. Due to the ambiguity, a phrase in a query may be mistakenly linked to a certain entity in the knowledge base. Thus, the semantic features include some noises. However, even with noises such features can still contribute to QS. To illustrate how the semantic features work, consider the query with the assigned semantic label as follows,

[history of the]<sub>NULL</sub> [search engine]<sub>computer software genre</sub>

As pointed out by (Tan et al., 2008), QS approaches which are only based on statistical information (e.g. *MI* and frequencies of n-grams) collected from the Web, cannot guarantee that the resulting segments are meaningful ones. For the query ‘history of the search engine’, a possible segmentation is ‘history of the | search engine’, as both ‘history of the’ and ‘search engine’ occur on the web frequently. In contrast, semantic information can distinguish ‘search engine’ from ‘history of the’, as ‘history of the’ is labeled as NULL and ‘search engine’ is labeled as ‘computer software genre’. Moreover, the learning algorithm can also learn some implicit relations between transformations and semantic labels, e.g. some particular combination of labels for  $s_1$ ,  $s_2$  and  $s_3$  may often trigger or prevent a transformation.

**Rank, Direction and Position Features.** Table 2 shows the values of these features. The rank feature is designed to distinguish among the different segmentation rankings of a base segmenter. For example, this feature can capture the intuition that for a good base segmenter, top ranked segmentations should have more of a chance to be selected. The direction feature is used to distinguish the two kinds of transformations: ‘ $w_iw_{i+1} \mapsto w_i|w_{i+1}$ ’ and ‘ $w_i|w_{i+1} \mapsto w_iw_{i+1}$ ’. The position feature considers decision positions, as transformations in different positions may have different chances.

Rank	$j$ , the rank of the segmentation to which we transform the top-1 segmentation.
Direction	1, if “ $w_iw_{i+1} \mapsto w_i w_{i+1}$ ”; 0, reverse.
Position <sup>left</sup>	Number of words from the decision position to the beginning/end of query.
Position <sup>right</sup>	

Table 2: The ‘rank’, ‘direction’ and ‘position’ features

## 5 Experiments

### 5.1 Experimental Setup

Following Hagen et al. (2011), we evaluate a QS system at three levels: **Query Level**:

$$Acc^{qry} = \frac{\#\text{correctly segmented queries}}{\#\text{queries in the evaluation data set}} \quad (5)$$

**Break Level.** The decision of break is whether or not to insert a segment boundary between two tokens in the query. The break-level accuracy ( $Acc^{brk}$ ) is defined as the proportion of the correctly-made decisions out of all such decisions.

**Segment Level.** Let  $Q^{eval}$  denote the set of queries.  $S_q^{sys}$  is the segmentation generated by a system and  $S_q^{eval}$  is given by a human. Then,

$$P^{sg} = \sum_{q \in Q^{eval}} \frac{|S_q^{sys} \cap S_q^{eval}|}{|S_q^{sys}|} \quad (6)$$

$$R^{sg} = \sum_{q \in Q^{eval}} \frac{|S_q^{sys} \cap S_q^{eval}|}{|S_q^{eval}|} \quad F^{sg} = \frac{2 \cdot P^{sg} \cdot R^{sg}}{P^{sg} + R^{sg}}$$

We use two data sets as introduced in (Bergsma and Wang, 2007) and (Hagen et al., 2010), denoted as ‘Bergsma-Wang-07’ (**BW07**) and ‘Webis-QSeC-10’ (**WQ10**). BW07 includes 500 test queries which all were noun phrase queries. Each query was segmented manually by three annotators (denoted as annotator A, B, and C) respectively. For 44% of the queries, all three annotators agree on the segmentations. Such an agreement between annotators cannot be considered as ‘strong’, which to some extent implies that human annotations may not be so reliable when used for training a segmentation model capable of consistently working over different queries. WQ10 includes 4,850 queries. Each query can be any type of query, not necessarily a noun phrase query. Each query was annotated by ten annotators.

We made use of the mining method in (Hu et al., 2011) for collecting the *query intent sets*. With the search log data and clicks (Apr 1, 2009-Mar 31, 2010) as input, we finally obtained 9,412,308 query intent sets, which totally include 30,902,284 unique queries. The similar queries in each set share more than 10 clicks. Each set includes 2~11 queries. We denote this data set as **QSet**. Note

that this data set does not have any annotations. Thus, we also tried to construct another data set (denoted as  $\mathbf{QSet}^{ann}$ ) by intersecting  $\mathbf{QSet}$  with WQ10.  $\mathbf{QSet}^{ann}$  includes 1,554 queries. Every query in  $\mathbf{QSet}^{ann}$  is then associated with the human annotations from WQ10 and linked to at least one query intent set in  $\mathbf{QSet}$ .

As each query has more than one segmentation due to different annotators, we select segmentation as our reference under two schemes: ‘**Majority**’ where the segmentations agreed upon by a majority of the annotators are chosen as the reference, and ‘**Best**’ where the annotated segmentations that maximize the break accuracy  $Acc^{brk}$  of the evaluated segmenter are chosen as the reference.

We mainly utilized three unsupervised systems as base segmenters. They are described in (Hagen et al., 2010), (Hagen et al., 2011) and (Risvik et al., 2003), denoted as  $\mathbf{Base}^{H-1}$ ,  $\mathbf{Base}^{H-2}$  and  $\mathbf{Base}^{CN}$  respectively. They can represent the state-of-the-art QS performance. For example,  $\mathbf{Base}^{H-2}$  on on data BW07(A) achieves 69.2%  $F^{sg}$  which slightly outperforms the recent unsupervised system (Li et al., 2011) (69.0%  $F^{sg}$ ).

As we focus on web QS, we did not compare LRMC with supervised methods which are only designed for one particular domain. For example, Yu et al. (2009)’s method is for queries of relational databases. The work (Bergsma and Wang, 2007) and the supervised stage of (Bendersky et al., 2009) are only for noun-phrases.

## 5.2 Consistency as Weak Supervision

LRMC relies on a training data which is automatically collected with the help of query intent sets. Thus, in this section, we evaluate the training set collected by  $\mathbf{M1}$  and  $\mathbf{M2}$  (Section 4.2).

In the experiments, we first applied a base segmenter to the queries in  $\mathbf{QSet}$  and then managed to choose one segmentation as correct from the output for each query with either  $\mathbf{M1}$  or  $\mathbf{M2}$ . Last, we evaluated the new output by checking only the segmentations for the queries in subset  $\mathbf{QSet}^{ann}$ . Some queries in  $\mathbf{QSet}^{ann}$  may belong to different intent-sets and in each intent-set may have different segmentation labels as ‘correct’. In our evaluation, we randomly selected one of them as the final label. Besides, we also included an ideal method **Oracle** by which the correct segmentation can always be identified and used as the new output if the segmentation exists in the top- $n$  re-

sults of the base segmenter. Note that **Oracle** is an upper-bound result obtained by directly matching with human’s annotation and cannot be applied to query intent sets  $\mathbf{QSet}$  for collecting labeled data. Table 3 provides the results, where top- $k$  ( $1 \leq k \leq 5$ ) means that the input to  $\mathbf{M1}/\mathbf{M2}$  is the top- $k$  results of the corresponding base segmenter. Note that the top-1 results are the performance of the base segmenters.

By checking the results of **Oracle**, we can find that for every base segmenter, more than 35% of the correct segmentations in the top-5 results are not covered by the top-1 results (in terms of  $Acc^{qry}$ ). In addition, around 90% correct segmentation boundaries ( $Acc^{brk}$ ) are included in the top-5 results of the base segmenters. These findings indicate the feasibility of our replacement model, which tries to replace a rank-1 segmentation by a rank- $k$  ( $k > 1$ ) segmentation.

From the table, we also see that both  $\mathbf{M1}$  and  $\mathbf{M2}$  are able to significantly perform better than the base segmenters do ( $p < 0.05$ , t-test). This can be observed through all the measures. For example, using  $Acc^{qry}$  as the evaluation metric, the percentage of the correct segmentations that  $\mathbf{M2}$  discovers more than the base segmenters do ranges from 20.2% to 24.6%. These improvements prove the underlying assumption that queries with a similar intent tend to have consistent segmentation. Besides, we can see that  $\mathbf{M2}$  can reach a satisfied performance to collect the labeled data. For example, break-level accuracy  $Acc^{brk}$  can reach 80%.

Table 3 also shows that  $\mathbf{M1}$  and  $\mathbf{M2}$  perform best by using top 3 or 4 results from base segmenter. This finding indicates that our framework should work with a reasonable base segmenter.

By comparing the results generated by  $\mathbf{M1}$  and  $\mathbf{M2}$  with all three measures, we see that  $\mathbf{M2}$  consistently performs better than  $\mathbf{M1}$ , and the difference is statistical significant ( $p < 0.05$ , t-test). This tells us that consistency should be calculated with all the top- $n$  segmentations rather than with only the selected segmentations.

## 5.3 Query Segmentation

In this section, we investigate the effectiveness of our LRMC which is a combination of data collecting method and replacement model.

In the experiments, we first made use of  $\mathbf{M2}$  to automatically collect the training data from the query intent sets  $\mathbf{QSet}$ . During the process of col-

Segmenter	Rank	Oracle			M1			M2		
		$Acc^{qry}$	$Acc^{brk}$	$F^{sg}$	$Acc^{qry}$	$Acc^{brk}$	$F^{sg}$	$Acc^{qry}$	$Acc^{brk}$	$F^{sg}$
Base <sup>CN</sup>	Top-1	38.7	67.9	51.7	38.7	67.9	51.7	38.7	67.9	51.7
	Top-2	46.1	76.1	60.8	42.4	70.0	55.6	47.0	76.4	56.3
	Top-3	67.3	85.6	74.9	<b>58.8</b>	<b>81.3</b>	<b>68.3</b>	58.8	81.0	68.1
	Top-4	73.3	88.5	79.9	58.3	79.6	67.6	<b>58.9</b>	<b>81.4</b>	<b>68.3</b>
	Top-5	<b>75.2</b>	<b>89.5</b>	<b>81.5</b>	44.6	72.1	57.6	59.6	61.0	60.2
Base <sup>H-1</sup>	Top-1	42.9	69.7	53.8	42.9	69.7	53.8	42.9	69.7	53.8
	Top-2	59.0	81.7	68.9	46.0	75.9	60.6	47.2	77.2	61.7
	Top-3	72.5	87.8	78.5	63.2	83.0	70.8	<b>65.3</b>	82.5	<b>72.0</b>
	Top-4	75.2	89.4	81.2	<b>64.3</b>	<b>83.3</b>	<b>71.6</b>	64.3	<b>83.5</b>	71.8
	Top-5	<b>77.9</b>	<b>90.6</b>	<b>83.2</b>	59.0	81.7	68.7	63.0	82.1	70.0
Base <sup>H-2</sup>	Top-1	39.6	68.3	52.2	39.6	68.3	52.2	39.6	68.3	52.2
	Top-2	51.6	78.5	64.2	45.6	74.0	59.4	45.4	73.5	59.2
	Top-3	69.4	86.4	76.3	<b>61.0</b>	<b>80.1</b>	<b>69.4</b>	64.2	<b>83.4</b>	<b>71.2</b>
	Top-4	74.1	88.9	80.5	59.2	78.9	69.0	63.1	82.2	70.0
	Top-5	<b>76.7</b>	<b>90.1</b>	<b>82.5</b>	59.8	78.0	67.2	<b>67.1</b>	66.3	66.5

Table 3: Consistency as weak supervision on QSet<sup>ann</sup> (Majority)

lecting the data, we took only the top-3 segmentations as input. The training data set collected by M2 contains around 45 million instances (pairs of segmentations). Then, all this labeled data is used to train the replacement model as introduced in Section 4.3. We made use of LIBSVM (Chang and Lin, 2011) and a linear kernel in our experiment. Finally, we applied the learned replacement models to the evaluation data sets BW07 and WQ10.

Table 4 reports the QS results<sup>1</sup>. Following previous work (Bergsma and Wang, 2007; Hagen et al., 2011), we report four groups of results with the data BW07. In each of the first three groups, only the reference segmentations from annotator A, B or C are used. The fourth group is ‘Best’ of BW07. We also report two groups of results (‘Majority’ and ‘Best’) with the data WQ10. Comparing each pair of ‘Base’ and ‘LRMC’, we can see that LRMC proposed in this paper can be successfully spliced onto different base segmenters and significantly improves them over different data sets under the three evaluation metrics  $Acc^{qry}$ ,  $Acc^{brk}$  and  $F^{sg}$ . ( $p < 0.05$ , **t-test**). Especially, the state-of-the-art systems Base<sup>H-1</sup> and Base<sup>H-2</sup> have been significantly improved by LRMC. The improvements prove that the automatically-collected labeled data can guide QS and our replacement model can take advantage of the data.

## 6 Conclusions and Future Work

We have proposed a paradigm LRMC for QS. LRMC assumes the existence of a base segmenter

<sup>1</sup>Note that the results for the base segmenters Base<sup>H-1</sup> and Base<sup>H-2</sup> are not exactly same as those reported in (Hagen et al., 2011) although they are very close. For example, Base<sup>H-1</sup> and Base<sup>H-2</sup> on WQ10 achieve 73.4%  $F^{sg}$  and 74.2%  $F^{sg}$  in the original paper. Ours are 71.2% and 72.1%. The reasons are as follows: For BW07, they used a cleaned version of the data set; for WQ10, they released just a subset of the data used in their experiments.

Data Set	Measure	Base <sup>CN</sup>		Base <sup>H-1</sup>		Base <sup>H-2</sup>	
		Base	LRMC	Base	LRMC	Base	LRMC
BW07 (A)	$Acc^{qry}$	53.4	55.3	55.2	56.7	53.8	55.4
	$Acc^{brk}$	79.3	81.4	80.2	81.9	79.5	81.7
	$F^{sg}$	66.5	69.6	67.5	70.2	66.7	69.8
BW07 (B)	$Acc^{qry}$	37.4	40.2	39.8	41.1	37.8	39.8
	$Acc^{brk}$	73.7	74.9	74.7	76.4	73.8	75.4
	$F^{sg}$	54.3	58.3	55.6	58.7	54.5	58.1
BW07 (C)	$Acc^{qry}$	41.6	44.2	43.8	46.7	42.0	46.9
	$Acc^{brk}$	74.1	75.2	75.0	78.6	74.2	78.6
	$F^{sg}$	56.9	60.4	58.0	62.3	57.1	62.4
BW07 (Best)	$Acc^{qry}$	62.2	67.2	64.6	66.2	65.6	66.8
	$Acc^{brk}$	85.1	90.0	86.1	87.3	87.6	88.7
	$F^{sg}$	74.5	79.6	75.8	78.4	78.6	79.5
WQ10 (Majority)	$Acc^{qry}$	30.0	38.5	31.8	40.1	30.3	39.8
	$Acc^{brk}$	65.3	72.1	66.2	74.0	65.5	73.1
	$F^{sg}$	47.5	55.1	48.5	55.8	47.7	55.6
WQ10 (Best)	$Acc^{qry}$	52.8	60.4	57.0	67.9	59.1	67.6
	$Acc^{brk}$	80.5	84.7	83.5	89.6	84.4	89.5
	$F^{sg}$	67.8	72.7	71.2	79.0	72.1	78.8

Table 4: Performance on query segmentation

and then learns how to select correct segmentations from the output of the base segmenter. The replacement model is trained by a labeled data set which can be automatically collected from *query intent sets*, instead of relying on any human annotation. There exist two interesting directions for future work: (1) we observe that there is still a big gap in performance between the proposed methods and Oracle. According to our analysis, most of the gap is caused by that the incorrect segmentations for some similar queries also happen to have a high consistency when measured by either proposed strategy. Thus, it is worth studying other methods that can address such performance gap. (2) we would like to further explore the concept of query intent sets. In this paper, we assume that similar intent queries tend to have similar segmentations. A reasonable next step is to explore the idea that similar intent queries tend to have similar labels, which can be useful for the task of tagging query segments with semantic labels.



## References

- A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *COLING*, 1998.
- M. Bendersky, W. B. Croft, and D. A. Smith. Two-stage query segmentation for information retrieval. In *SIGIR*, 2009.
- S. Bergsma and Q. I. Wang. Learning noun phrase query segmentation. In *EMNLP-CoNLL*, 2007.
- K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 2008.
- C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. *Intelligent Systems and Technology*, 2011.
- C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *TREC*, 2009.
- J. Guo, G. Xu, H. Li, and X. Cheng. A unified and discriminative model for query refinement. In *SIGIR*, 2008.
- M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. The power of naive query segmentation. In *SIGIR*, 2010.
- M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In *WWW*, 2011.
- Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng. Mining query subtopics from search log data. In *WWW*, 2011.
- J. Huang, J. Gao, J. Miao, X. Li, K. Wang, F. Behr, and C. L. Giles. Exploring web scale language models for search query processing. In *WWW*, 2010.
- R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW*, 2006.
- J. Kiseleva, Q. Guo, E. Agichtein, D. Billsus, and W. Chai. Unsupervised query segmentation using click data: preliminary results. In *WWW*, 2010.
- Y. Li, B.-J. P. Hsu, C. Zhai, and K. Wang. Unsupervised query segmentation using clickthrough for information retrieval. In *SIGIR*, 2011.
- N. Mishra, R. S. Roy, N. Ganguly, S. Laxman, and M. Choudhury. Unsupervised query segmentation using only query logs. In *WWW*, 2011.
- K. M. Risvik, T. Mikolajewski, and P. Boros. Query segmentation for web search. In *WWW*, 2003.
- D. Roth and D. Zelenko. Part of speech tagging using a network of linear separators. In *Coling-Acl, The 17th International Conference on Computational Linguistics*, 1998.
- B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *SIGIR*, 2010.
- B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. In *WWW*, 2008.
- D. Wu, Y. Zhang, and T. Liu. Unsupervised query segmentation using monolingual word alignment method. *Comp. and Info. Science*, 5(1), 2012.
- X. Yu and H. Shi. Query segmentation using conditional random fields. In *KEYS*, 2009.
- C. Zhang, N. Sun, X. Hu, T. Huang, and T.-S. Chua. Query segmentation based on eigenspace similarity. In *ACL-IJCNLP*, 2009.

# Precise Information Retrieval Exploiting Predicate-Argument Structures

Daisuke Kawahara<sup>†</sup> Keiji Shinzato<sup>‡</sup> Tomohide Shibata<sup>†</sup> Sadao Kurohashi<sup>†</sup>

<sup>†</sup>Graduate School of Informatics, Kyoto University

<sup>‡</sup>Rakuten Institute of Technology

{dk, shibata, kuro}@i.kyoto-u.ac.jp, keiji.shinzato@mail.rakuten.com

## Abstract

A concept can be linguistically expressed in various syntactic constructions. Such syntactic variations spoil the effectiveness of incorporating dependencies between words into information retrieval systems. This paper presents an information retrieval method for normalizing syntactic variations via predicate-argument structures. We conduct experiments on standard test collections and show the effectiveness of our approach. Our proposed method significantly outperforms a baseline method based on word dependencies.

## 1 Introduction

Most conventional approaches to information retrieval (IR) deal with words as independent terms. In query sentences<sup>1</sup> and documents, however, dependencies exist between words.<sup>2</sup> To capture these dependencies, some extended IR models have been proposed in the last decade (Jones, 1999; Lee et al., 2006; Song et al., 2008; Shinzato et al., 2008). These models, however, did not achieve consistent significant improvements over models based on independent words.

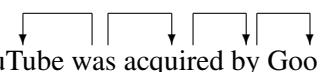
One of the reasons for this is the linguistic variations of syntax, that is, languages are syntactically expressed in various ways. For instance, the same or similar meaning can be expressed using the passive voice or the active voice in a sentence. Previous approaches based on dependencies cannot identify such variations. This is because they use the output of a dependency parser, which generates syntactic (grammatical) dependencies built

<sup>1</sup>In this paper, we handle queries written in natural language.

<sup>2</sup>While dependencies between words are sometimes considered to be the co-occurrence of words in a sentence, in this paper we consider dependencies to be syntactic or semantic dependencies between words.

upon surface word sequences. Consider, for example, the following sentence in a document:

(1) YouTube was acquired by Google.



Dependency parsers based on the Penn Treebank and the head percolation table (Collins, 1999) judge the head of “YouTube” as “was” (“YouTube←was”; hereafter, we denote a dependency by “*modifier←head*”). This dependency, however, cannot be matched with the dependency “YouTube←acquire” in a query like:

(2) I want to know the details of the news that Google acquired YouTube.

Furthermore, even if a dependency link in a query matches that in a document, a mismatch of dependency type can cause another problem. This is because previous models did not distinguish dependency types. For example, the dependency “YouTube←acquire” in query sentence (2) can be found in the following irrelevant document.

(3) Google acquired PushLife for \$25M ...  
YouTube acquired Green Parrot Pictures ...

While this document does indeed contain the dependency “YouTube←acquire,” its type is different; specifically, the query dependency is accusative while the document dependency is nominative. That is to say, ignoring differences in dependency types can lead to inaccurate information retrieval.

In this paper, we propose an IR method that does not use syntactic dependencies, but rather predicate-argument structures, which are normalized forms of sentence meanings. For example, query sentence (2) is interpreted as the following predicate-argument structure (hereafter, we denote a predicate-argument structure by  $\langle \cdot \cdot \cdot \rangle$ ):<sup>3</sup>

<sup>3</sup>In this paper, we use the following abbreviations:

(4) ⟨NOM:Google acquire ACC:YouTube⟩.

Sentence (1) is also represented as the same predicate-argument structure, and documents including this sentence can be regarded as relevant documents. Conversely, the irrelevant document (3) has different predicate-argument structures from (4), as follows:

(5) a. ⟨NOM:Google acquire ACC:PushLife⟩,

b. ⟨NOM:YouTube acquire ACC:Green Parrot Pictures⟩.

In this way, by considering this kind of predicate-argument structure, more precise information retrieval is possible.

We mainly evaluate our proposed method using the NTCIR test collection, which consists of approximately 11 million Japanese web documents. We also have an experiment on the TREC Robust 2004 test collection, which consists of around half a million English documents, to validate the applicability to other languages than Japanese.

This paper is organized as follows. Section 2 introduces related work, and section 3 describes our proposed method. Section 4 presents the experimental results and discussion. Section 5 describes the conclusions.

## 2 Related work

There have been two streams of related work that considers dependencies between words in a query sentence.

One stream is based on linguistically-motivated approaches that exploit natural language analysis to identify dependencies between words. For example, Jones proposed an information retrieval method that exploits linguistically-motivated analysis, especially dependency relations (Jones, 1999). However, Jones noted that dependency relations did not contribute to significantly improving performance due to the low accuracy and robustness of syntactic parsers. Subsequently, both the accuracy and robustness of dependency parsers were dramatically improved (Nivre and Scholz, 2004; McDonald et al., 2005), with such parsers being applied more recently to information retrieval (Lee et al., 2006; Song et al., 2008; Shin-

NOM (nominative), ACC (accusative), DAT (dative), ALL (allative), GEN (genitive), CMI (comitative), LOC (locative), ABL (ablative), CMP (comparative), DEL (delimitative) and TOP (topic marker).

zato et al., 2008). For example, Shinzato et al. investigated the use of syntactic dependency output by a dependency parser and reported a slight improvement over a baseline method that used only words. However, the use of dependency parsers still introduces the problems stated in the previous section because of their handling of only syntactic dependencies.

The second stream of research has attempted to integrate dependencies between words into information retrieval models. These models include a dependence language model (Gao et al., 2004), a Markov Random Field model (Metzler and Croft, 2005), and a quasi-synchronous dependence model (Park et al., 2011). However, they focus on integrating term dependencies into their respective models without explicitly considering any syntactic or semantic structures in language. Therefore, the purpose of these studies can be considered different from ours.

Park and Croft (2010) proposed a method for ranking query terms for the selection of those which were most effective by exploiting typed dependencies in the analysis of query sentences. They did not, however, use typed dependencies for indexing documents.

The work that is closest to our present work is that of Miyao et al. (2006), which proposed a method for the semantic retrieval of relational concepts in the domain of biomedicine. They retrieved sentences that match a given query using predicate-argument structures via a framework of region algebra. Thus, they namely approached the task of sentence matching, which is not the same as document retrieval (or ranking). As for the types of queries they used, although their method could handle natural language queries, they used short queries like “TNF activate IL6.” Because of the heavy computational load of region algebra, if a query matches several thousand sentences, for example, then it requires several thousand seconds to return all sentence matches (though it takes on average 0.01 second to return the first matched sentence).

In the area of question answering, predicate-argument structures have been used to precisely match a query with a passage in a document (e.g., (Narayanan and Harabagiu, 2004; Shen and Lapata, 2007; Bilotti et al., 2010)). However, candidate documents to extract an answer are retrieved using conventional search engines without

predicate-argument structures.

### 3 Information retrieval exploiting predicate-argument structures

#### 3.1 Overview

Our key idea is to exploit the normalization of linguistic expressions based on their predicate-argument structures to improve information retrieval.

The process of information retrieval systems can be decomposed into offline processing and online processing. During offline processing, analysis is first applied to a document collection. For example, typical analyses for English include tokenization and stemming analyses, while those for Japanese include morphological analysis. In addition, previous models using the dependencies between words also used dependency parsing. In this paper, we employ predicate-argument structures analysis, which is detailed in the next subsection.

Following the initial analysis, indexing is performed to produce an inverted index. In most cases, words are indexed as terms, but several previous approaches have also indexed dependencies between words as terms (e.g., (Shinzato et al., 2008)). In our study, however, we do not use syntactic dependencies directly, but rather consider predicate-argument structures. To bring this predicate-argument structure information into the index, we handle predicate-argument structures as a set of typed semantic dependencies. Dependency types are expressed as term features, which are additional information to each term including the list of positions of the term.

As for online processing, we first apply the predicate-argument structure analysis to a query sentence, and then create terms including words and typed semantic dependencies extracted from the predicate-argument structures. Then, we search documents containing these terms from the inverted index, and then finally rank these documents.

In the following subsections, we describe in more detail the procedures of predicate-argument structure analysis, indexing, query processing, and document ranking.

#### 3.2 Analysis of predicate-argument structures

We apply predicate-argument structure analysis to both queries and documents. Predicate-argument

structure analysis normalizes the following linguistic expressions:

- relative clause
- passive voice (the predicate is normalized to active voice)
- causative (the predicate is normalized to normal form)
- intransitive (the predicate is normalized to transitive)
- giving and receiving expressions (the predicate is normalized to a giving expression)

In the case of Japanese, we use the morphological analyzer JUMAN,<sup>4</sup> and the predicate-argument structure analyzer KNP (Kawahara and Kurohashi, 2006).<sup>5</sup> The accuracy of syntactic dependencies output by KNP is around 89% and that of predicate-argument relations is around 81% on web sentences. Examples of this predicate-argument structure analysis are shown in Figures 1 and 2. Figure 1 shows an example of relative clause normalization by predicate-argument structure analysis. The syntactic dependencies of the two sentences are different, but this difference is solved by using predicate-argument structures.

Figure 2 shows an example of intransitive verb normalization by predicate-argument structure analysis. In this example, the syntactic dependencies are the same, but different verbs are used.<sup>6</sup> The analyzer canonicalizes the intransitive verb to its corresponding transitive verb, and also produces the same predicate-argument structure for the two sentences.

If we apply our method to English, deep parsers such as the Stanford Parser<sup>7</sup> and Enju<sup>8</sup> can be employed to achieve predicate-argument structure analysis. The Stanford parser can output typed semantic dependencies that conform to the Stanford dependencies (de Marneffe et al., 2006). Enju is an HPSG parser that outputs predicate-argument structures, and arguments are typed as Arg1, Arg2, and so forth. The representation of the dependency types in Enju is the same as that of PropBank (Palmer et al., 2005).

<sup>4</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

<sup>5</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP>

<sup>6</sup>In many cases, the lemma of a transitive verb is not the same as that of its corresponding intransitive verb in Japanese.

<sup>7</sup><http://www-nlp.stanford.edu/software/lex-parser.shtml>

<sup>8</sup><http://www.nactem.ac.uk/enju/>

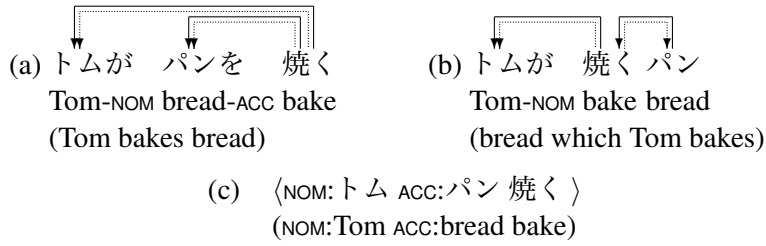


Figure 1: An example of relative clause normalization by predicate-argument structure analysis in Japanese. (a) is a normal-order sentence and (b) is a sentence that contains a relative clause, “トムが焼く” (which Tom bakes). Arrows represent syntactic dependencies. Dotted arrows represent semantic dependencies that constitute predicate-argument structures. Both sentences are normalized to the predicate-argument structure (c).

In this way, though our framework itself is language-independent, our method depends on the availability of a predicate-argument structure analyzer for the target language.

### 3.3 Indexing

Our method builds an inverted index from the results of the predicate-argument structure analysis. First, word lemmas are registered as terms. We then need to integrate the predicate-argument structure information into the index. One possibility is to represent each predicate-argument structure as a term, but this method leads to a data sparseness problem. This is because the number of arguments in predicate-argument structures varies greatly not only in documents, but also in queries because of information granularity. For example, to express the same event, a predicate-argument structure can omit time or place information.

Instead, we decompose a predicate-argument structure into a set of typed semantic dependencies. A typed semantic dependency is defined as a typed dependency between a predicate and an argument that the predicate governs. For instance, the predicate-argument structure in Figure 2 can be decomposed into the following two typed semantic dependencies:

- (6) a. トム  $\xleftarrow{\text{NOM}}$  上げる  
(Tom  $\xleftarrow{\text{NOM}}$  raise)
- b. テンション  $\xleftarrow{\text{ACC}}$  上げる  
(tension  $\xleftarrow{\text{ACC}}$  raise)

These typed semantic dependencies are registered as dependency terms in the index. The type information is encoded as a *term feature*, which is an additional field for each dependency term. This term feature consists of both dependency type information and predicate information. We con-

sider major postpositions in Japanese as dependency types (Table 1). If a dependency type is not listed in this table, then this type is regarded as a special type which we classify as “other.” In addition, a dependency that is not the relation between a predicate and its argument is also classified as “other” (e.g., the dependency between verbs).

The predicate information in the term feature refers to the original predicate type for canonicalized predicates. There are four types: passive, causative, intransitive, and giving expression.

### 3.4 Query processing

Hereafter, we describe the steps of online processing. When a query sentence is input, both predicate-argument structure analysis and term extraction are applied to the query sentence in the same way indexing is applied. The extracted terms consist of words and typed semantic dependencies and they are used to retrieve documents.

Note that unnecessary expressions like “教えてください” (please tell me) in a query sentence are not used to extract terms.

### 3.5 Document retrieval and scoring

Using the results of the query processing, documents are then retrieved and ranked. First, documents are retrieved by accessing the inverted index using the terms extracted from the query analysis. Here, we have two options for the logical operator on the terms. If we apply the logical operator AND, we impose a constraint that all the terms must be contained in a retrieved document. Conversely, if we apply the logical operator OR, a retrieved document should have one of the terms. In this study, we use the logical operator OR to retrieve as many documents as possible. This means that we do not apply any methods of selecting or

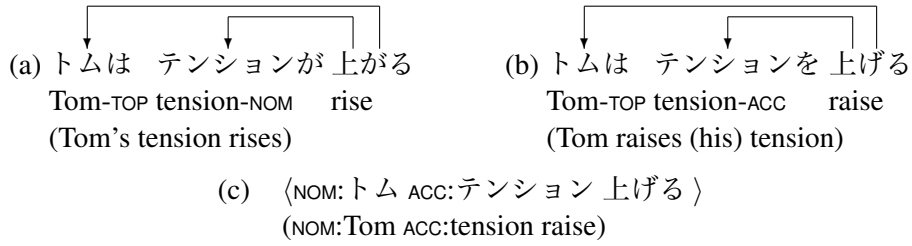


Figure 2: An example of intransitive verb normalization by predicate-argument structure analysis in Japanese. (a) is an intransitive sentence and (b) is a transitive sentence. Arrows represent syntactic dependencies (they are also semantic dependencies in this case). Both sentences are normalized to the predicate-argument structure (c). In particular, the intransitive verb “上がる” (rise) is a different word from the transitive verb “上げる” (raise) but both are canonicalized to the same transitive verb “上げる” (raise) in the predicate-argument structure.

が	を	に	と	で	から	まで	より	時間	修飾	の	について	として
NOM	ACC	DAT	CNJ	LOC	ABL	DEL	CMP	time	<i>adj</i>	GEN	about	as

Table 1: Dependency type information in Japanese. The first row is the list of dependency types used in our method. The second row means the translations of the first row, where *adj* means adjuncts such as adverbs.

ranking query terms,<sup>9</sup> but rely only on document scoring to examine the effectiveness of the use of predicate-argument structures.

Following document retrieval, a relevancy score is assigned to each document, and the documents are ranked according to these relevancy scores. We use Okapi BM25 (Robertson et al., 1992) for estimating the relevancy score between a query and a document. This measure was originally proposed for models based on terms of independent words, but we slightly extend this measure to include estimating relevancy for typed semantic dependencies that are extracted from predicate-argument structures. Our relevancy score is calculated as a weighted sum of the score of words and the score of dependencies. The score of dependencies is further calculated as a weighted sum of the following two scores: the score of dependencies with consistent (matched) type and that with inconsistent (mismatched) type. In particular, the score of dependencies with inconsistent type is reduced compared to the score of dependencies with consistent type.

We denote a set of words in a query  $q$  as  $T_{qw}$ , and also denote a set of dependencies in  $q$  as  $T_{qd}$ . This set of dependencies is further divided into two types according to the consistency of dependency features:  $T_{qd_C}$  (consistent) and  $T_{qd_I}$  (inconsistent). We define the relevancy score between

query  $q$  and document  $d$  as follows:

$$R(q, d) = \sum_{t \in T_{qw}} BM(t, d) + \beta \left\{ \sum_{t \in T_{qd_C}} BM(t, d) + \gamma \sum_{t \in T_{qd_I}} BM(t, d) \right\}, \quad (1)$$

where  $\beta$  is a parameter for adjusting the ratio of a score calculated from dependency relations to that from words and  $\gamma$  is a parameter for decreasing the weight of inconsistent dependency types. The score  $BM(t, d)$  is defined as:

$$BM(t, d) = IDF(t) \times \frac{(k_1+1)F_{dt}}{K+F_{dt}} \times \frac{(k_3+1)F_{qt}}{k_3+F_{qt}}, \quad (2)$$

$$IDF(t) = \log \frac{N-n+0.5}{n+0.5},$$

$$K = k_1 \left\{ (1-b) + b \frac{l_d}{l_{ave}} \right\},$$

where  $F_{dt}$  is the frequency with which  $t$  appears in document  $d$ ,  $F_{qt}$  is the frequency that  $t$  appears in  $q$ ,  $N$  is the number of documents being searched,  $n$  is the document frequency of  $t$ ,  $l_d$  is the length of document  $d$  (words), and  $l_{ave}$  is the average document length. Finally, we set these Okapi parameters as  $k_1 = 1$ ,  $k_3 = 0$  and  $b = 0.6$ .

We use the following relevancy score for a baseline method that uses only syntactic dependencies, which is explained in section 4:

$$R(q, d) = \sum_{t \in T_{qw}} BM(t, d) + \beta \sum_{t \in T_{qd}} BM(t, d). \quad (3)$$

<sup>9</sup>We only discard unnecessary expressions in a query as described in subsection 3.4.

This equation is the same as the relevancy score used in Shinzato et al. (2008).

## 4 Evaluation

In this section, we evaluate and analyze our proposed method on the standard test collections of Japanese and English.

### 4.1 Evaluation on Japanese Test Collection

#### 4.1.1 Experimental setup

We implemented our proposed method using the open search engine infrastructure TSUBAKI (Shinzato et al., 2008) as a base system. TSUBAKI generates an inverted index from linguistic analyses in an XML format. Note that while TSUBAKI has a facility for using a synonym lexicon, but we did not use it because we performed pure comparisons without referencing synonyms.

We evaluated our proposed method by using the test collection built for the NTCIR-3 (Eguchi et al., 2003) and NTCIR-4 (Eguchi et al., 2004) workshops. These workshops shared a target document set, which consists of 11,038,720 web pages from Japanese domains. We used a high-performance computing environment to perform predicate-argument structure analysis and indexing on these documents. It took three days for analysis and two days for indexing. For the evaluation, we used 127 informational topics (descriptions) defined in the test collections (47 from NTCIR-3 and 80 from NTCIR-4). We also had additional 65 topics that were not used for evaluation in NTCIR-3; we used these 65 topics for parameter tuning. The relevance of each document with respect to a topic was judged as highly relevant, relevant, partially relevant, irrelevant or unjudged. We regarded the highly relevant, relevant, and partially relevant documents as correct answers.

For each topic, we retrieved 1,000 documents, ranked according to the score  $R(q, d)$  in equation (1). We optimized the parameter  $\beta$  as 0.18, and the parameter  $\gamma$  as 0.85 using the additional 65 topics in relation to their mean average precision (MAP) score. We then assessed retrieval performance according to MAP, P@3 (Precision at 3), P@5, P@10 and nDCG@10 (Järvelin and Kekäläinen, 2002). Note that unjudged documents were treated as irrelevant when computing the scores. For the graded relevance of nDCG@10, we mapped highly relevant, relevant, and partially relevant to the values 3, 2, and 1, respectively.

	MAP	P@3	P@5	P@10	nDCG@10
word	0.1665	0.4233	0.4159	0.3706	0.2323
word+dep	0.1704	0.4233	0.4095	0.3730	0.2313
word+pa	<b>0.1727**</b>	<b>0.4418*</b>	<b>0.4175</b>	<b>0.3794*</b>	<b>0.2370**</b>

Table 2: Retrieval performance of two baseline methods (“word” and “word+dep”) and our proposed method (“word+pa”). \*\* and \* mean that the differences between “word+dep” and “word+pa” are statistically significant with  $p < 0.05$  and  $p < 0.10$ , respectively.

	MAP	P@3	P@5	P@10	nDCG@10
word	0.2085	0.4312	0.4302	0.3960	0.2455
word+dep	0.2120	0.4392	0.4286	0.3913	0.2433
word+pa	<b>0.2139**</b>	<b>0.4524</b>	<b>0.4333</b>	<b>0.3976**</b>	<b>0.2484**</b>

Table 3: Retrieval performance without unjudged documents. \*\* means that the differences between “word+dep” and “word+pa” are statistically significant with  $p < 0.05$ .

#### 4.1.2 Retrieval performance evaluation

Table 2 lists retrieval performances. In this table, “word” is a baseline method that uses only words as terms, and “word+dep” is another baseline method that uses words and untyped syntactic dependencies as terms. These untyped syntactic dependencies are also available in the results of the predicate-argument structure analyzer KNP. “word+pa” is our proposed model, which considers predicate-argument structures. We also applied the Wilcoxon signed-rank test to the differences between “word+dep” and “word+pa.”

We can see that our proposed method “word+pa” outperformed the baselines “word” and “word+dep” in all the metrics. In particular, the difference between “word+dep” and “word+pa” in MAP was statistically significant with  $p = 0.01134$ . In addition, P@3 is higher than the baselines by approximately 1.9%. This means that our model can provide more relevant documents on the top of the ranked result. The baseline “word+dep” outperformed the baseline “word” in MAP, which is used as a metric for optimizing the parameters, but did not outperform “word” in P@5 and nDCG@10. That is to say, “word+dep” was not consistently better than “word.”

Generally, relevance judgments on a standard test collection are created using a pooling method, which judges a certain number of documents submitted by every participating system. Systems that are developed after the creation of the test col-

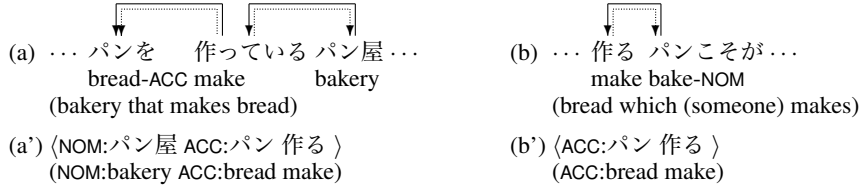


Figure 3: An improved example of relative clause normalization by predicate-argument structure analysis in Japanese. (a) is a part of the query sentence and (b) is a part of a relevant document. Arrows represent syntactic dependencies and dotted arrows represent semantic dependencies. These sentences are normalized to the predicate-argument structures (a') and (b'), respectively.

	MAP	P@3	P@5	P@10	nDCG@10
word+dep	0.1769	0.4444	0.4254	0.3921	0.2373
word+pa	<b>0.1790**</b>	<b>0.4577</b>	<b>0.4317</b>	<b>0.3984*</b>	<b>0.2424**</b>

Table 4: Retrieval performance including additional judgments. The meaning of \*\* and \* is the same as the previous tables.

lection possibly retrieve unjudged documents, but they are usually handled as irrelevant documents, even though they may contain relevant documents. In addition, the number of unjudged documents is likely to increase according to the complexity of systems. To alleviate this bias, we evaluated the three systems without the inclusion of unjudged documents. Table 3 lists the evaluation results. From this table, we can see that “word” was likely to defeat “word+dep,” but “word+pa” consistently outperformed the two baseline methods.

We also evaluated unjudged documents manually. We asked a person who is a certified librarian to judge them. These documents comprise the unjudged documents which appeared in the top 10 results of the two methods (“word+dep” and “word+pa”) for each topic. Table 4 lists the retrieval performances reflecting the inclusion of these additional judgments. From this table, the result of proposed method is consistently better than that of the baseline using syntactic dependencies.

### 4.1.3 Discussions

By introducing the normalization by predicate-argument structures, our proposed method can retrieve relevant documents that cannot be retrieved or ranked below 1,000 documents by the baseline methods. Figures 3 and 4 show improved examples by the proposed method (“word+pa”) compared to the baseline method (“word+dep”). Figure 3 is an example of the effect of normalizing relative clauses. The following sentences are the original query and a part of relevant document:

- (7) a. 天然酵母のパンを作っているパン屋を見つけない  
(I want to find shops that make bread with natural yeast.)  
b. ...塩、酵母のみで作るパンこそが、...  
(... only the bread that (someone) makes using only salt and yeast ...)

Here, (a) is a query and (b) is a sentence in a relevant document. These sentences have different syntactic dependencies as illustrated in Figure 3, but they are normalized to the predicate-argument structures (a') and (b') in Figure 3. The whole predicate-argument structures are different, but they contain the same typed semantic dependency:

- (8) パン  $\xleftarrow{\text{ACC}}$  作る  
(bread  $\xleftarrow{\text{ACC}}$  make).

Figure 4 is an example of the effect of normalizing intransitive verbs. The following sentences are the original sentences in a query and a relevant document:

- (9) a. 各地域でお正月に食べる雑煮に入っている具、またはベースとなる味噌などの違いについて調べたい  
(I wish to find out about differences in the ingredients and miso stock used to make ozoni soup at New Years in each region.)  
b. ... 北海道のお雑煮はシャケやイクラ、じゃがいもを入れるところもある  
(in some places, they put salmon, salmon roe and potato in ozoni soup in Hokkaido)

While different verbs are used to express almost the same meaning in these sentences, they are normalized to the predicate-argument structures (a') and (b') in Figure 4. The whole predicate-argument structures are different, but they contain the same typed semantic dependency:



<p>(a) ... 雑煮に 入っている 具 ...  ozoni-DAT exist ingredients  (ingredients that exist in ozoni soup)</p> <p>(a') &lt;ACC:具 DAT:雑煮 入れる &gt;  (ACC:ingredients DAT:ozoni soup put)</p>	<p>(b) 雑煮は ... 入れる ...  ozoni-TOP put  (put in ozoni soup)</p> <p>(b') &lt;DAT:雑煮 入れる &gt;  (DAT:ozoni soup put)</p>
---	--

Figure 4: An improved example of intransitive verb normalization by predicate-argument structure analysis in Japanese. (a) is a part of the query sentence and (b) is a part of a relevant document. These sentences are normalized to the predicate-argument structures (a') and (b'), respectively. In particular, the intransitive verb “入る” (exist) is a different word from the transitive verb “入れる” (put) but both are canonicalized to the same transitive verb “入れる” (put) in the predicate-argument structures.

(10) 雑煮  $\xleftarrow{\text{DAT}}$  入れる  
(ozoni soup  $\xleftarrow{\text{DAT}}$  put).

Generally speaking, linguistic variations can be roughly divided into two types: syntactic variations and lexical variations. Among syntactic variations, we handled syntactic variations that are related to predicate-argument structures in this study. In our future work, we intend to investigate remaining syntactic variations, such as nominal compounds and paraphrases consisting of larger trees than predicate-argument structures.

The other type is lexical variations, namely synonymous words and phrases. In our approach, they are partially handled in the normalization process to predicate-argument structures. Although handling lexical variations is not the main focus of this paper, we will investigate the effect of incorporating a lexicon of synonymous words and phrases into our model.

## 4.2 Evaluation on English Test Collection

To validate the effectiveness of the proposed method in other languages than Japanese, we also conducted an experiment on English. We used the TREC Robust 2004 test collection (Voorhees, 2004), which consists of 528,155 English documents and 250 topics (TREC topics 301-450 and 601-700). We used the description queries in these topics, which are written in natural language. Stopwords are removed from the parse of a description and dependencies that contain a stopword in either a modifier or a head are also removed. We used the INQUERY stopword list (Allan et al., 2000). Other experimental settings are the same as the Japanese evaluation.

Table 5 lists retrieval performances. In this table, “word” is a baseline method that uses only lemmatized words as terms, and “word+dep” is another baseline method that uses lemmatized words and syntactic dependencies that are analyzed by the state-of-the-art dependency parser

	MAP	P@3	P@5	P@10	nDCG@10
word	0.1344	0.4498	0.4016	0.3297	0.3527
word+dep	0.1350	0.4337	0.4112	0.3317	0.3517
word+pa	<b>0.1396*</b>	<b>0.4618**</b>	<b>0.4257**</b>	<b>0.3482**</b>	<b>0.3659**</b>

Table 5: Retrieval performance of two baseline methods (“word” and “word+dep”) and our proposed method (“word+pa”) on the TREC test collection. The meaning of \*\* and \* is the same as the previous tables.

MaltParser.<sup>10</sup> “word+pa” is our proposed method, which considers predicate-argument structures converted from the typed semantic dependencies output by the Stanford Parser.<sup>11</sup> We can see that our proposed method “word+pa” outperformed the baselines “word” and “word+dep” in all the metrics also on this English test collection.

## 5 Conclusions

This paper described an information retrieval method that exploits predicate-argument structures to precisely capture the dependencies between words. Experiments on the standard test collections of Japanese and English indicated the effectiveness of our approach. In particular, the proposed method outperformed a baseline method that uses syntactic dependencies output by a dependency parser.

For future work, we plan to optimize ranking by using machine learning techniques such as support vector regression, and to capture any remaining syntactic differences that express similar meanings (i.e., paraphrasing). We used the Okapi BM25 system as our baseline in this study. We will also employ a language model-based information retrieval system as a baseline to confirm the robustness of our approach.

<sup>10</sup><http://www.maltparser.org/>

<sup>11</sup>To normalize passive constructions, we applied a rule that converts the dependency type “nsubjpass” to “dobj” and “agent” to “nsubj.”

## Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 23680015.

## References

- James Allan, Margaret E. Connell, W. Bruce Croft, Fangfang Feng, David Fisher, and Xiaoyan Li. 2000. INQUERY and TREC-9. In *Proceedings of the Ninth Text REtrieval Conference*, pages 551–562.
- Matthew W Bilotti, Jonathan Elsas, Jaime Carbonell, and Eric Nyberg. 2010. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of CIKM2010*, pages 459–468. ACM.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *the 5th International Conference on Language Resources and Evaluation*.
- Koji Eguchi, Keizo Oyama, Emi Ishida, Noriko Kando, and Kazuko Kuriyama. 2003. The web retrieval task and its evaluation in the third NTCIR workshop. In *Proceedings of SIGIR2003*.
- Koji Eguchi, Keizo Oyama, Akiko Aizawa, and Haruko Ishikawa. 2004. Overview of web task at the fourth NTCIR workshop. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization*.
- Jianfeng Gao, Jian-Yun Nie, Guanyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of SIGIR2004*, pages 170–177.
- Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.
- Karen Sparck Jones. 1999. What is the role of NLP in text retrieval? In T. Strzalkowski, editor, *Natural language information retrieval*, pages 1–24. Kluwer.
- Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of HLT-NAACL2006*, pages 176–183.
- Changki Lee, Gary Geunbae Lee, and Myung-Gil Jang. 2006. Dependency structure applied to language modeling for information retrieval. *ETRI Journal*, 28(3):337–346.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL2005*, pages 91–98.
- Donald Metzler and W. Bruce Croft. 2005. A markov random field model for term dependencies. In *Proceedings of SIGIR2005*, pages 472–479.
- Yusuke Miyao, Tomoko Ohta, Katsuya Masuda, Yoshimasa Tsuruoka, Kazuhiro Yoshida, Takashi Nomiya, and Jun’ichi Tsujii. 2006. Semantic retrieval for the accurate identification of relational concepts in massive textbases. In *Proceedings of COLING-ACL2006*, pages 1017–1024.
- Srini Narayanan and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of COLING2004*, pages 184–191.
- Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING2004*, pages 64–70.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Jae-Hyun Park and W. Bruce Croft. 2010. Query term ranking based on dependency parsing of verbose queries. In *Proceedings of SIGIR2010*, pages 829–830.
- Jae-Hyun Park, W. Bruce Croft, and David A. Smith. 2011. Quasi-synchronous dependence model for information retrieval. In *Proceedings of CIKM2011*, pages 17–26.
- Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. 1992. Okapi at TREC. In *Proceedings of Text REtrieval Conference*, pages 21–30.
- Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of EMNLP-CoNLL2007*, pages 12–21.
- Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. TSUBAKI: An open search engine infrastructure for developing new information access methodology. In *Proceedings of IJCNLP2008*, pages 189–196.
- Young-In Song, Kyoung-Soo Han, Sang-Bum Kim, So-Young Park, and Hae-Chang Rim. 2008. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems*, 31(3):265–286.
- Ellen M. Voorhees. 2004. Overview of the TREC 2004 robust retrieval track. In *Proceedings of Text REtrieval Conference 2004*.

# Global Model for Hierarchical Multi-Label Text Classification

Yugo Murawaki

Graduate School of Informatics  
Kyoto University  
murawaki@i.kyoto-u.ac.jp

## Abstract

The main challenge in hierarchical multi-label text classification is how to leverage hierarchically organized labels. In this paper, we propose to exploit dependencies among multiple labels to be output, which has been left unused in previous studies. To do this, we first formalize this task as a structured prediction problem and propose (1) a global model that jointly outputs multiple labels and (2) a decoding algorithm for it that finds an exact solution with dynamic programming. We then introduce features that capture inter-label dependencies. Experiments show that these features improve performance while reducing the model size.

## 1 Introduction

Hierarchical organization of a large collection of data has deep roots in human history (Berlin, 1992). The emergence of electronically-available text has enabled us to take computational approaches to real-world hierarchical text classification tasks. Such text collections include patents,<sup>1</sup> medical taxonomies<sup>2</sup> and Web directories such as Yahoo! and the Open Directory Project.<sup>3</sup> In this paper, we focus on *multi-label* classification, in which a document may be given more than one label.

Hierarchical multi-label text classification is a challenging task because it typically involves thousands of labels and an exponential number of output candidates. For efficiency, divide-and-conquer strategies have often been adopted. Typically, the label hierarchy is mapped to a set of local

<sup>1</sup><http://www.wipo.int/classifications/en/>

<sup>2</sup><http://www.nlm.nih.gov/mesh/>

<sup>3</sup><http://www.dmoz.org/>

classifiers, which are invoked in a top-down fashion (Montejo-Ráez and Ureña-López, 2006; Wang et al., 2011; Sasaki and Weissenbacher, 2012). However, local search is difficult to harness because a chain of local decisions often leads to what is usually called error propagation (Bennett and Nguyen, 2009). To alleviate this problem, previous work has resorted to what we collectively call post-training adjustment.

One characteristic of the task that has not been explored in previous studies is that multiple labels to be output have dependencies among them. It is difficult even for human annotators to decide how many labels they choose. We conjecture that they consult the label hierarchy when adjusting the number of output labels. For example, if two label candidates are positioned proximally in the hierarchy, human annotators may drop one of them because they provide overlapping information.

In this paper, we propose to exploit inter-label dependencies. To do this, we first formulate hierarchical multi-label text classification as a structured prediction problem. We propose a global model that jointly predicts a set of labels. Under this framework, we replace local search with dynamic programming to find an exact solution. This allows us to extend the model with features for inter-label dependencies. Instead of locally training a set of classifiers, we also propose global training to find globally optimal parameters. Experiments show that these features improve performance while reducing the model size.

## 2 Task Definition

In hierarchical multi-label text classification, our goal is to assign to a document a set of labels  $\mathbf{m} \subset \mathcal{L}$  that best represents the document. The pre-defined set of labels  $\mathcal{L}$  is organized as a tree as illustrated in Figure 1.<sup>4</sup> In our task, only the

<sup>4</sup>Some studies work on directed acyclic graphs (DAGs), in which each node can have more than one parent (Labrou and

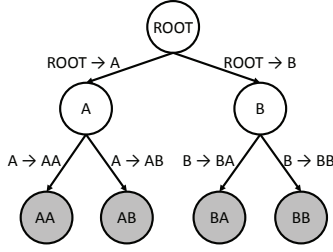


Figure 1: Example of label hierarchy. Leaf nodes, filled in gray, represent labels to be assigned to documents.

leaf nodes (AA, AB, BA and BB in this example) represent valid labels.

Let  $\text{leaves}(c)$  be a set of the descendants of  $c$ , inclusive of  $c$ , that are leaf nodes. For example,  $\text{leaves}(A) = \{AA, AB\}$ .  $p \rightarrow c$  denotes an edge from parent  $p$  to child  $c$ . Let  $\text{path}(c)$  be a set of edges that connect ROOT to  $c$ . For example,  $\text{path}(AB) = \{\text{ROOT} \rightarrow A, A \rightarrow AB\}$ . Let  $\text{tree}(\mathbf{m}) = \bigcup_{l \in \mathbf{m}} \text{path}(l)$ . It corresponds to a subtree that covers  $\mathbf{m}$ . For example,  $\text{tree}(\{AA, AB\}) = \{\text{ROOT} \rightarrow A, A \rightarrow AA, A \rightarrow AB\}$ .

We assume that each document  $x$  is transformed into a feature vector by  $\phi(x)$ . For example, we can use a bag-of-words representation of  $x$ .

We consider a supervised setting. The training data  $\mathcal{T} = \{(x_i, \mathbf{m}_i)\}_{i=1}^T$  is used to train our models. Their performance is measured on test data.

### 3 Base Models

#### 3.1 Flat Model

We begin with the flat model, one of the simplest models in multi-label text classification. It ignores the label hierarchy and relies on a set of binary classifiers, each of which decides whether label  $l$  is to be assigned to document  $x$ .

Various models have been used to implement binary classifiers, including Naïve Bayes, Logistic Regression and Support Vector Machines. We use the Perceptron family of algorithms, and it will be extended later to handle more complex structures.

The binary classifier for label  $l$  is associated with a weight vector  $\mathbf{w}_l$ . If  $\mathbf{w}_l \cdot \phi(x) > 0$ , then  $l$  is assigned to  $x$ . Note that at least one label is assigned to  $x$ . If no labels have positive scores, we choose one label with the highest score.

To optimize  $\mathbf{w}_l$ , we convert the original training

Finin, 1999; LSHTC3, 2012). We leave it for future work.

---

**Algorithm 1** Passive-Aggressive algorithm for training a binary classifier (PA-I).

---

**Input:** training data  $\mathcal{T}_l = \{(x_i, y_i)\}_{i=1}^T$   
**Output:** weight vector  $\mathbf{w}_l$

- 1:  $\mathbf{w}_l \leftarrow \mathbf{0}$
- 2: **for**  $n = 1..N$  **do**
- 3:   shuffle  $\mathcal{T}_l$
- 4:   **for all**  $(x, y) \in \mathcal{T}_l$  **do**
- 5:      $l \leftarrow \max\{0, 1 - y(\mathbf{w}_l \cdot \phi(x))\}$
- 6:     **if**  $l > 0$  **then**
- 7:        $\tau \leftarrow \min\{C, \frac{l}{\|\phi(x)\|^2}\}$
- 8:        $\mathbf{w}_l \leftarrow \mathbf{w}_l + \tau y \phi(x)$
- 9:     **end if**
- 10:   **end for**
- 11: **end for**

---

data  $\mathcal{T}$  into  $\mathcal{T}_l$ .

$$\mathcal{T}_l = \left\{ (x_i, y_i) \mid \begin{array}{l} y_i = +1 \text{ if } l \in \mathbf{m}_i \\ y_i = -1 \text{ otherwise} \end{array} \right\}_{i=1}^T$$

Each document is treated as a positive example if it has label  $l$ ; otherwise it is a negative example. Since local classifiers are independent of each other, we can trivially parallelize training.

We employ the Passive-Aggressive algorithm for training (Crammer et al., 2006). Specifically we use PA-I. The pseudo-code is given in Algorithm 1. We set the aggressiveness parameter  $C$  as 1.0.

#### 3.2 Tree Model

Unlike the flat model, the tree model exploits the label hierarchy. Each local classifier is now associated with an edge  $p \rightarrow c$  of the label hierarchy and has a weight vector  $\mathbf{w}_{p \rightarrow c}$ . If  $\mathbf{w}_{p \rightarrow c} \cdot \phi(x) > 0$ , it means that  $x$  would belong to descendant(s) of  $c$ . Edge classifiers are independent of each other and can be trained in parallel.

We consider two ways of constructing training data  $\mathcal{T}_{p \rightarrow c}$ .

**ALL** — All training data are used as before.

$$\mathcal{T}_{p \rightarrow c} = \left\{ (x_i, y_i) \mid \begin{array}{l} y_i = +1 \text{ if } \exists l \in \mathbf{m}_i, l \in \text{leaves}(c) \\ y_i = -1 \text{ otherwise} \end{array} \right\}_{i=1}^T$$

Each document is treated as a positive example if it belongs to a leaf node of  $c$ , and the rest is negative examples (Punera and Ghosh, 2008).

**SIB** — Negative examples are restricted documents that belong to the leaves of  $c$ 's siblings.

$$\mathcal{T}_{p \rightarrow c} = \left\{ (x, y) \mid \begin{array}{l} y = +1 \text{ if } \exists l \in \mathbf{m}, l \in \text{leaves}(c) \\ y = -1 \text{ if } \exists l \in \mathbf{m}, l \in \text{leaves}(p) \\ \text{and } l \notin \text{leaves}(c) \end{array} \right\}$$

---

**Algorithm 2** Top-down local search.

---

**Input:** document  $x$   
**Output:** label set  $\mathbf{m}$

```
1:  $q \leftarrow [\text{ROOT}], \mathbf{m} \leftarrow \{\}$ 
2: while  $q$  is not empty do
3:    $p \leftarrow$  pop out the first item of  $q, \mathbf{t} \leftarrow \{\}$ 
4:   for all  $c$  such that  $c$  is a child of  $p$  do
5:      $\mathbf{t} \leftarrow \mathbf{t} \cup \{(c, \mathbf{w}_{p \rightarrow c} \cdot \phi(x))\}$ 
6:   end for
7:    $\mathbf{u} \leftarrow \{(c, s) \in \mathbf{t} \mid s > 0\}$ 
8:   if  $\mathbf{u}$  is empty then
9:      $\mathbf{u} \leftarrow \{(c, s)\}$  such that  $c$  has the highest score  $s$ 
       among  $p$ 's children
10:  end if
11:  for all  $(c, s) \in \mathbf{u}$  do
12:    if  $c$  is a leaf node then
13:       $\mathbf{m} \leftarrow \mathbf{m} \cup \{c\}$ 
14:    else
15:      append  $c$  to  $q$ 
16:    end if
17:  end for
18: end while
```

---

This leads to a compact model because low-level edges, which are overwhelming in number, have much smaller training data than high-level edges. This is a preferred choice in previous studies (Liu et al., 2005; Wang et al., 2011; Sasaki and Weissenbacher, 2012).

### 3.3 Top-down Local Search

In previous studies, the tree model is usually accompanied with top-down local search for decoding (Montejo-Ráez and Ureña-López, 2006; Wang et al., 2011; Sasaki and Weissenbacher, 2012).<sup>5</sup> Algorithm 2 is a basic form of top-down local search. At each node, we select children to which edge classifiers return positive scores (Lines 4–7). However, if no children have positive scores, we select one child with the highest score (Lines 8–10). We repeat this until we reach leaves. The decoding of the flat model can be seen as a special case of this search.

Top-down local search is greedy, hierarchical pruning. If a higher-level classifier drops a child node, we no longer consider its descendants as output candidates. This drastically reduces the number of local classifications in comparison with the flat model. At the same time, however, this is a source of errors. In fact, a chain of local decisions accumulates errors, which is known as error propagation (Bennett and Nguyen, 2009). If the decision by a higher-level classifier was wrong, the model has no way of recovering from the error.

---

<sup>5</sup>For other methods, Punera and Ghosh (2008) post-process local classifier outputs by isotonic tree regression.

To alleviate this problem, various modifications have been proposed, which we collectively call post-training adjustment. Sasaki and Weissenbacher (2012) combined broader candidate generation with post-hoc pruning. They first generated a larger number of candidates by setting a negative threshold (e.g.,  $-0.2$ ) instead of 0 in Line 7. Then they filtered out unlikely labels by setting another threshold on the sum of (sigmoid-transformed) local scores of each candidate's path. S-cut (Montejo-Ráez and Ureña-López, 2006; Wang et al., 2011) adjusts the threshold for each classifier. R-cut selects top- $r$  candidates either globally (Liu et al., 2005; Montejo-Ráez and Ureña-López, 2006) or at each parent node (Wang et al., 2011). Wang et al. (2011) developed a meta-classifier which classified a root-to-leaf path using sigmoid-transformed local scores and some additional features. All these methods assume that the models themselves are inherently imperfect and must be supplemented by additional parameters which are tuned manually or by using development data.

## 4 Proposed Method

### 4.1 Global Model

We see hierarchical multi-label text classification as a structured prediction problem. We propose a global model that jointly predicts  $\mathbf{m}$ , or  $\text{tree}(\mathbf{m})$ .

$$\text{score}(x, \mathbf{m}) = \mathbf{w} \cdot \Phi(x, \text{tree}(\mathbf{m}))$$

$\mathbf{w}$  can be constructed simply by combining local edge classifiers.

$$\mathbf{w} = \mathbf{w}_{\text{ROOT} \rightarrow A} \oplus \mathbf{w}_{\text{ROOT} \rightarrow B}, \dots, \oplus \mathbf{w}_{B \rightarrow BB}$$

Its corresponding feature function  $\Phi(x, \text{tree}(\mathbf{m}))$  returns copies of  $\phi(x)$ , each of which corresponds to an edge of the label hierarchy. Thus  $\text{score}(x, \mathbf{m})$  can be reformulated as follows.

$$\text{score}(x, \mathbf{m}) = \sum_{p \rightarrow c \in \text{tree}(\mathbf{m})} \mathbf{w}_{p \rightarrow c} \cdot \phi(x)$$

Now we want to find  $\mathbf{m}$  that maximizes the global score,  $\arg\max_{\mathbf{m}} \text{score}(x, \mathbf{m})$ .

With the global model, we can confirm that local search is a major source of errors. In preliminary experiments, we trained local edge classifiers on **ALL** data and combined the resultant classifiers to create a global model. For 33% of documents in the same dataset, local search found sets of labels whose global scores were lower than the corresponding correct sets of labels.

---

**Algorithm 3** MAXTREE( $x, p$ )

---

**Input:** document  $x$ , tree node  $p$ **Output:** label set  $\mathbf{m}$ , score  $s$ 

```

1:  $\mathbf{u} \leftarrow \{\}$ 
2: for all  $c$  in the children of  $p$  do
3:   if  $c$  is a leaf then
4:      $\mathbf{u} \leftarrow \mathbf{u} \cup \{(\{c\}, \mathbf{w}_{p \rightarrow c} \cdot \phi(x))\}$ 
5:   else
6:      $(\mathbf{m}', s') \leftarrow \text{MAXTREE}(x, c)$ 
7:      $\mathbf{u} \leftarrow \mathbf{u} \cup \{(\mathbf{m}', s' + \mathbf{w}_{p \rightarrow c} \cdot \phi(x))\}$ 
8:   end if
9: end for
10:  $\mathbf{r} \leftarrow \{(\mathbf{m}, s) \in \mathbf{u} \mid s > 0\}$ 
11: if  $\mathbf{r}$  is empty then
12:    $\mathbf{r} \leftarrow \{(\mathbf{m}, s)\}$  such that the item has the highest score
      $s$  among  $\mathbf{u}$ 
13: end if
14:  $\mathbf{m} \leftarrow \bigcup_{(\mathbf{m}, s) \in \mathbf{r}} \mathbf{m}$ 
15:  $s \leftarrow \sum_{(\mathbf{m}, s) \in \mathbf{r}} s$ 
16: return  $(\mathbf{m}, s)$ 

```

---

## 4.2 Dynamic Programming

We show that an exact solution for the global model can be found by dynamic programming.<sup>6</sup> The pseudo-code is given in Algorithm 3. MAXTREE( $x, p$ ) recursively finds a subtree that maximizes the score rooted by  $p$ , and thus we invoke MAXTREE( $x, \text{ROOT}$ ). For  $p$ , each child  $c$  is associated with (1) a set of labels that maximizes the score of the subtree rooted by  $c$  and (2) its score (Lines 3–8). The score of  $c$  is the sum of  $c$ 's tree score and the score of the edge  $p \rightarrow c$ . A leaf's tree score is zero.

To maximize  $p$ 's tree score, we select all children that add positive scores to the parent (Line 10). If no children add positive scores, we select one child that gives the highest score (Lines 11–13). Again, the flat model can be seen as a special case of this algorithm. The selected children correspond to  $p$ 's label set and score (Lines 14–15).

A possible extension to this algorithm is to output  $N$ -best label sets. Since our algorithm is much easier than bottom-up parsing (McDonald et al., 2005), it would not be so difficult (Collins and Koo, 2005).

Dynamic programming resolves the search problem. We no longer require post-training adjustment. It allows us to concentrate on improving the model itself.

<sup>6</sup>Bennett and Nguyen (2009) proposed a similar method, but neither global model nor global training was considered. In their method, the scores of lower-level classifiers were incorporated as meta-features of a higher-level classifier. All these classifiers were trained locally and required burdensome cross-validation techniques.

---

**Algorithm 4** Modification to incorporate branching features. Replace Lines 10–15 of Algorithm 3.

---

```

10:  $r \leftarrow \mathbf{u}$  sorted by  $s$  in descending order
11:  $\mathbf{r}' \leftarrow \{\}$ ,  $s' \leftarrow 0$ ,  $\mathbf{m}' \leftarrow \{\}$ 
12: for  $k = 1..size$  of  $r$  do
13:    $(\mathbf{m}, s) \leftarrow r[k]$ 
14:    $s' \leftarrow s' + s$ ,  $\mathbf{m}' \leftarrow \mathbf{m}' \cup \mathbf{m}$ 
15:    $\mathbf{r}' \leftarrow \mathbf{r}' \cup \{(\mathbf{m}', s' + \mathbf{w}_{\text{BF}} \cdot \phi_{\text{BF}}(p, k))\}$ 
16: end for
17:  $(\mathbf{m}, s) \leftarrow$  item in  $\mathbf{r}'$  that has the highest  $s$ 

```

---

## 4.3 Inter-label Dependencies

Now we are ready to exploit inter-label dependencies. We introduce branching features, a simple but powerful extension to the global model. They influence how many children a node selects. The corresponding function is  $\phi_{\text{BF}}(p, k)$ , where  $p$  is a non-leaf node and  $k$  is the number of children to be selected for  $p$ . To avoid sparsity, we choose one of  $R + 1$  features ( $1, \dots, R$  or  $>R$ ) for some pre-defined  $R$ . To be precise, we fire two features per non-leaf node: one is node-specific and the other is shared among non-leaf nodes. As a result, we append at most  $(I + 1)(R + 1)$  features to the global weight vector, where  $I$  is the number of non-leaf nodes.

All we have to do to incorporate branching features is to replace Lines 10–15 of Algorithm 3 with Algorithm 4. For given  $k$ , we first need to select  $k$  children that maximize the sum of the scores. This can be done by sorting children by score and select the first  $k$  children. We then add a score of branching features  $\mathbf{w}_{\text{BF}} \cdot \phi_{\text{BF}}(p, k)$  (Line 15). Finally we chose a candidate with the highest score (Line 17).

## 4.4 Global Training

Up to this point, the global model is constructed by combining locally trained classifiers. Of course, we can directly train the global model. In fact we cannot incorporate branching features without global training.

Algorithm 5 shows a Passive-Aggressive algorithm for the structured output (Crammer et al., 2006). We can find an exact solution under the current weight vector by dynamic programming (Line 5).<sup>7</sup> The cost  $\rho$  reflects the degree to which the model's prediction was wrong. It is based on the

<sup>7</sup>If we want for some reason to stick to local search, we need to address the problem of "non-violation." With inexact search, the model prediction  $\hat{\mathbf{m}}$  may have a lower score than correct  $\mathbf{m}$ , making the update invalid. Several methods have been proposed to solve this problem (Collins and Roark, 2004; Huang et al., 2012).

---

**Algorithm 5** Passive-Aggressive algorithm for global training (PA-I, prediction-based updates).

---

**Input:** training data  $\mathcal{T} = \{(x_i, \mathbf{m}_i)\}_{i=1}^T$   
**Output:** weight vector  $\mathbf{w}$

- 1:  $\mathbf{w} \leftarrow \mathbf{0}$
- 2: **for**  $n = 1..N$  **do**
- 3:   shuffle  $\mathcal{T}$
- 4:   **for all**  $(x, \mathbf{m}) \in \mathcal{T}$  **do**
- 5:     predict  $\hat{\mathbf{m}} \leftarrow \operatorname{argmax}_{\mathbf{m}} \operatorname{score}(x, \mathbf{m})$
- 6:      $\rho \leftarrow 1 - 2|\mathbf{m} \cap \hat{\mathbf{m}}|/(|\mathbf{m}| + |\hat{\mathbf{m}}|)$
- 7:     **if**  $\rho > 0$  **then**
- 8:        $l \leftarrow \operatorname{score}(x, \hat{\mathbf{m}}) - \operatorname{score}(x, \mathbf{m}) + \sqrt{\rho}$
- 9:        $\tau \leftarrow \min\{C, \frac{l}{\|\Phi(x, \operatorname{tree}(\mathbf{m})) - \Phi(x, \operatorname{tree}(\hat{\mathbf{m}))\|^2}\}$
- 10:        $\mathbf{w} \leftarrow \mathbf{w} + \tau(\Phi(x, \operatorname{tree}(\mathbf{m})) - \Phi(x, \operatorname{tree}(\hat{\mathbf{m}})))$
- 11:     **end if**
- 12:   **end for**
- 13: **end for**

---

example-based F measure, which will be reviewed in Section 5.3.

Note that what are called “global” in some previous studies are in fact *path*-based methods (Qiu et al., 2009; Qiu et al., 2011; Wang et al., 2011; Sasaki and Weissenbacher, 2012). In contrast, we present *tree-wide* optimization.

#### 4.5 Parallelization of Global Training

One problem with global training is speed. We can no longer train local classifiers in parallel because global training makes the model monolithic. Even worse, label set prediction is orders of magnitude slower than a binary classification. For these reasons, global training is extremely slow.

We resort to iterative parameter mixing (McDonald et al., 2010). The basic idea is to split training data into small “shards” instead of subdividing the model. Algorithm 6 gives a pseudocode, where  $S$  is the number of shards. We perform training on each shard in parallel. At the end of each iteration, we average the models and use the resultant model as the initial value for the next iteration.

Iterative parameter mixing was originally proposed for Perceptron training. However, as McDonald et al. (2010) noted, it is possible to provide theoretical guarantees for distributed online Passive-Aggressive learning.

## 5 Experiments

### 5.1 Dataset

We used JSTplus, a bibliographic database on science, technology and medicine built by Japan Science and Technology Agency (JST).<sup>8</sup> Each docu-

<sup>8</sup><http://www.jst.go.jp/EN/menu3/01.html>

---

**Algorithm 6** Iterative parameter mixing for global training.

---

**Input:** training data  $\mathcal{T} = \{(x_i, \mathbf{m}_i)\}_{i=1}^T$   
**Output:** weight vector  $\mathbf{w}$

- 1: split  $\mathcal{T}$  into  $\mathcal{S}_1, \dots, \mathcal{S}_S$
- 2:  $\mathbf{w} \leftarrow \mathbf{0}$
- 3: **for**  $n = 1..N$  **do**
- 4:   **for**  $s = 1..S$  **do**
- 5:      $\mathbf{w}_s \leftarrow$  asynchronously call Algorithm 5 with some modifications:  $\mathcal{T}$  is replaced with  $\mathcal{S}_s$ ,  $\mathbf{w}$  is initialized with  $\mathbf{w}$  instead of  $\mathbf{0}$ , and  $N$  is set as 1.
- 6:   **end for**
- 7:   join
- 8:    $\mathbf{w} \leftarrow \frac{1}{S} \sum_{s=1}^S \mathbf{w}_s$
- 9: **end for**

---

ment consisted of a title, an abstract, a list of authors, a journal name, a set of categories and many other fields. For experiments, we selected a set of documents that (1) were dated 2010 and (2) contained both Japanese title and abstract. As a result, we obtained 455,311 documents, which were split into 409,892 documents for training and 45,419 documents for evaluation.

The number of labels was 3,209, which amounts to 4,030 edges. All the leave nodes are located at the fifth level (the root not counted). Some edges skip intermediate levels (e.g., children of a second-level node are located at the fourth level). On average 1.85 categories were assigned to a document, with a variance of 0.85. The maximum number of categories per document was 9.

For the feature representation of a document  $\phi(x)$ , we employed two types of features.

1. Journal name (binary). One feature was fired per document.
2. Content words in the title and abstract (frequency-valued). Frequencies of the words in the title were multiplied by two.

To extract content words, we first applied the morphological analyzer JUMAN<sup>9</sup> to each sentence to segment it into a word sequence. From each word sequence, we selected content words using the dependency parser KNP,<sup>10</sup> which tagged content words at a pre-processing step. Each document contained 380 characters on average, which corresponded to 120 content words according to JUMAN and KNP.

<sup>9</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

<sup>10</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP>

## 5.2 Models

In addition to the flat model (**FLAT**), the tree model with various configurations was compared. We performed local training of edge classifiers on **ALL** data and **SIB** data as explained in Section 3.2. We applied top-down local search (**LS**) and dynamic programming (**DP**) for decoding. We also performed global training (**GT**) with and without branching features (**BF**).

We performed 10 iterations for training local classifiers. For iterative parameter mixing described in Section 4.5, we evenly split the training data into 10 shards and ran 10 iterations. For branching features introduced in Section 4.3, we set  $R = 3$ .

## 5.3 Evaluation Measures

Various evaluation measures have been proposed to handle multiple labels. The first group of evaluation measures we adopted is document-oriented measures often referred to as *example-based* measures (Godbole and Sarawagi, 2004; Tsoumakas et al., 2010). The example-based precision (**EBP**), recall (**EBR**) and F measure (**EBF**) are defined as follows.

$$\begin{aligned} \text{EBP} &= \frac{1}{T} \sum_{i=1}^T \frac{|\mathbf{m}_i \cap \hat{\mathbf{m}}_i|}{|\hat{\mathbf{m}}_i|} \\ \text{EBR} &= \frac{1}{T} \sum_{i=1}^T \frac{|\mathbf{m}_i \cap \hat{\mathbf{m}}_i|}{|\mathbf{m}_i|} \\ \text{EBF} &= \frac{1}{T} \sum_{i=1}^T \frac{2|\mathbf{m}_i \cap \hat{\mathbf{m}}_i|}{|\hat{\mathbf{m}}_i| + |\mathbf{m}_i|} \end{aligned}$$

where  $T$  is the number of documents in the test data,  $\mathbf{m}_i$  is a set of correct labels of the  $i$ -th document and  $\hat{\mathbf{m}}_i$  is a set of labels predicted by the model.

Another group of measures are called label-based (**LB**) and are based on the precision, recall and F measure of each label (Tsoumakas et al., 2010). Multiple label scores are combined by performing macro-averaging (**Ma**) or micro-averaging (**Mi**), resulting in six measures.

Lastly we used hierarchical evaluation measures to give some scores to “partially correct” labels (Kiritchenko, 2005). If we assume a tree instead of a more general directed acyclic graph, we can formulate the (micro-average) hierarchical

precision (**hP**) and hierarchical recall (**hR**) as follows.

$$\begin{aligned} \text{hP} &= \frac{\sum_{i=1}^T |\text{tree}(\mathbf{m}_i) \cap \text{tree}(\hat{\mathbf{m}}_i)|}{\sum_{i=1}^T |\text{tree}(\hat{\mathbf{m}}_i)|} \\ \text{hR} &= \frac{\sum_{i=1}^T |\text{tree}(\mathbf{m}_i) \cap \text{tree}(\hat{\mathbf{m}}_i)|}{\sum_{i=1}^T |\text{tree}(\mathbf{m}_i)|} \end{aligned}$$

The hierarchical F measure (hF) is the harmonic mean of hP and hR.

## 5.4 Results

Table 1 shows the performance comparison of various models. DP-GT-BF performed best in 5 measures. Compared with FLAT, DP-GT-BF drastically improved LBMiP and hP. Branching features consistently improved F measures. The tree model with local search was generally outperformed by the flat model. Compared with FLAT, DP-ALL and DP-GT, DP-GT-BF yielded statistically significant improvements with  $p < 0.01$ .

DP-ALL outperformed LS-ALL for all but one measures. DP-SIB performed extremely poorly while DP-ALL was competitive with DP-GT-BF. This is in sharp contrast to the pair of LS-ALL and LS-SIB, which performed similarly. Dynamic programming forced DP-SIB’s local classifiers to classify what were completely new to them because they had been trained only on small portions of data. The result was highly unpredictable.

As expected, dynamic programming was much slower than local search. In fact DP-GT-BF was more than 60 times slower than local search. Somewhat surprisingly, it took only 18% more time than FLAT. This may be explained by the fact that DP-GT-BF was 16% smaller in size than FLAT.

Although DP-ALL was competitive with DP-GT and DP-GT-BF, it is notable that global training yielded much smaller models. Branching features brought further model size reduction along with almost consistent performance improvement. This result seems to support our hypothesis concerning the decision-making process of the human annotators. They do not select each label independently but consider the relative importance among competing labels.

## 5.5 Discussion

Table 2 shows the performance of several models on the training data. It is interesting that FLAT and DP-ALL scored much higher on the training data



model	iterations	time (min)	size	EBP	EBR	EBF
FLAT	10	266	73M	.4520	.4111	.3956
LS-ALL	10	<b>5</b>	115M	.3927	.4064	.3713
LS-SIB	10	5	<b>39M</b>	.4010	.4396	.3881
DP-ALL	10	329	115M	.4790	.4336	.4247
DP-SIB	10	298	<b>39M</b>	.0026	<b>.6804</b>	.0481
DP-GT	10	310	68M	<b>.5177</b>	.4096	.4317
DP-GT-BF	10	315	62M	.5172	.4121	<b>.4347</b>

model	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF	hP	hR	hF
FLAT	.4260	.2549	.2578	.4155	.3727	.3930	.5343	.4746	.5027
LS-ALL	.3288	.2764	.2415	.3622	.3716	.3668	.4988	.5060	.5024
LS-SIB	.3291	.2989	.2515	.3415	.4066	.3712	.4750	.5359	.5036
DP-ALL	<b>.4576</b>	.2760	<b>.2799</b>	.4542	.3933	.4216	.6020	.5163	.5559
DP-SIB	.0267	<b>.5214</b>	.0406	.0184	<b>.6649</b>	.0358	.0031	<b>.8104</b>	.0600
DP-GT	.4301	.2708	.2659	.5085	.3655	.4253	.6458	.4843	.5535
DP-GT-BF	.4519	.2645	.2709	<b>.5132</b>	.3701	<b>.4300</b>	<b>.6493</b>	.4898	<b>.5584</b>

Table 1: Performance comparison of various models. Time is the one required to classify test data. Loading time was not counted. Size is defined as the number of elements in the weight vector whose absolute values are greater than  $10^{-7}$ .

model	EBF	LBMiF	hF
FLAT	.9227	.9204	.9337
DP-ALL	.8977	.8951	.9114
DP-SIB	.0731	.0540	.0743
DP-GT-BF	.7126	.6942	.7508

Table 2: Performance on the training data.

than DP-GT-BF although they were outperformed on the test data. It seems safe to conclude that local training caused overfitting.

We further investigated the models by decomposing them into edges. Figure 2 compares three models. The first three figures (a–c) report the number of non-trivial elements in each weight vector. Edges are grouped by the level of child nodes. Although DP-GT-BR was much smaller in total size than DP-ALL, the per-edge size distributions looked alike. The higher the level was, the larger number of non-trivial features each model required. Compared with DP-SIB, DP-GT-BR had compact local classifiers for the highest-level edges but the rest was generally larger. Intuitively, knowing its siblings is not enough for each local classifier, but it does not need to know all possible rivals.

The last three figures (d–f) report the averaged absolute scores of each edge that were calculated from the model output for the test data. By doing

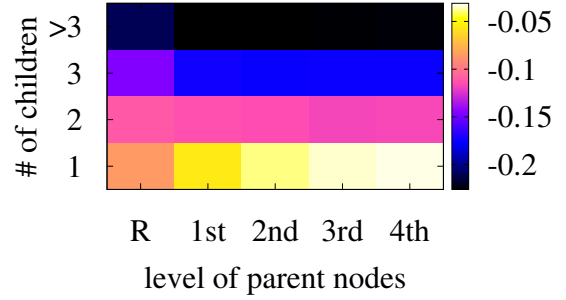


Figure 3: Heat map of the weight vector for branching features. R is the root level.

this, we would like to measure how edges of various levels affect the model output. Higher-level edges tended to have larger impact. However, we can see that in DP-GT-BR, their impact was relatively small. In other words, lower-level edges played more important roles in DP-GT-BR than in other models.

Figure 3 shows a heat map representation of the weight vector for DP-GT-BR’s branching features. The value of each item is the weight value averaged over parent nodes. All averaged weight values were negative. The penalty monotonically increased with the number of children. It is not easy to compare different levels of nodes because weight values depended on other parts of the weight vector. However, the fact that lower-level nodes marked sharper contrasts between small and large number of children appears to support our

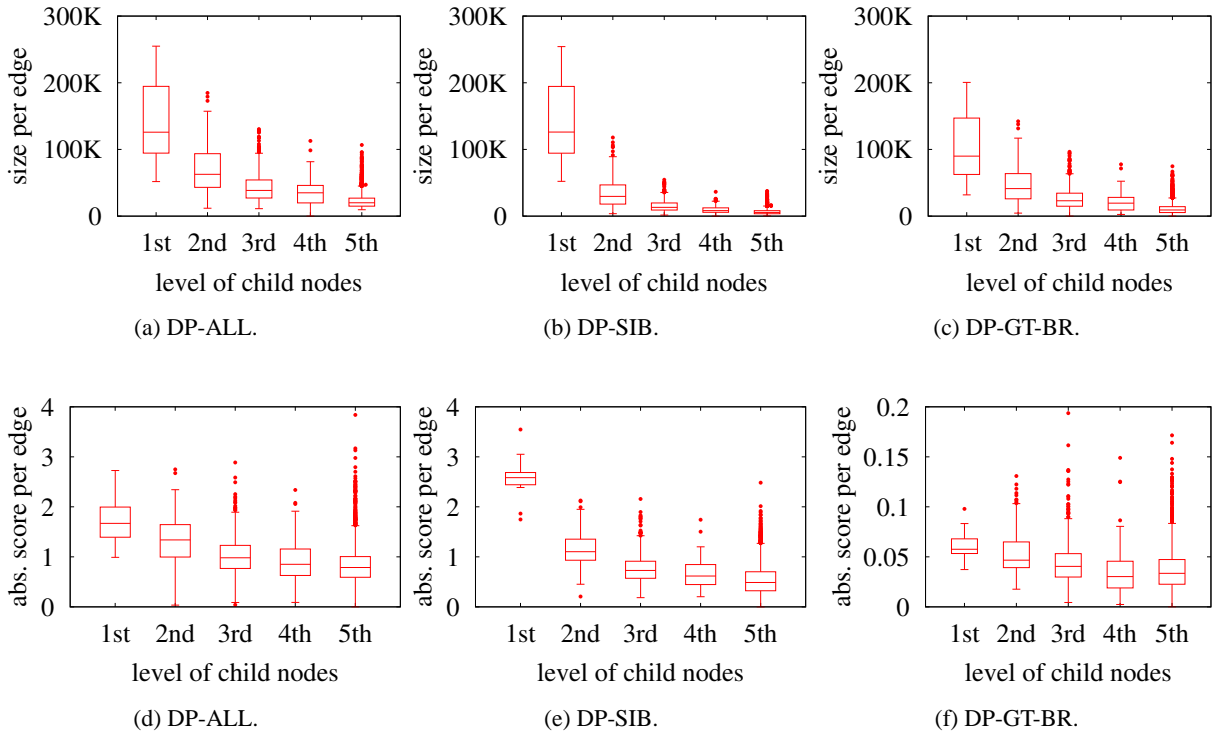


Figure 2: Comparison of model sizes and scores per edge. The definition of size is the same as that in Table 1.

hypothesis about the competitive nature of label candidates positioned proximally in the label hierarchy.

## 6 Conclusion

In this paper, we treated hierarchical multi-label text classification as a structured prediction problem. Under this framework, we proposed (1) dynamic programming that finds an exact solution, (2) global training and (3) branching features that capture inter-label dependencies. Branching features improve performance while reducing the model size. This result suggests that the selection of multiple labels by human annotators greatly depends on the relative importance among competing labels.

Exploring features that capture other types of inter-label dependencies is a good research direction. For example, “Others” labels probably behave atypically in relation to their siblings. While we focus on the setting where only the leaf nodes represent valid labels, internal nodes are sometimes used as valid labels. Such internal nodes often block the selection of their descendants. Also, we would like to work on directed acyclic graphs and to improve scalability in the future.

## Acknowledgments

We thank the Department of Databases for Information and Knowledge Infrastructure, Japan Science and Technology Agency for providing JST-Plus and helping us understand the database. This work was partly supported by JST CREST.

## References

- Paul N. Bennett and Nam Nguyen. 2009. Refined experts: improving classification in large taxonomies. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '09*, pages 11–18.
- Brent Berlin. 1992. *Ethnobiological classification: principles of categorization of plants and animals in traditional societies*. Princeton University Press.
- Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.
- 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.
- 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.
- Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative methods for multi-labeled classification. In Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, editors, *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *Lecture Notes in Computer Science*, pages 22–30. Springer Berlin Heidelberg.
- 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.
- Svetlana Kiritchenko. 2005. *Hierarchical Text Categorization and Its Application to Bioinformatics*. Ph.D. thesis, University of Ottawa.
- Yannis Labrou and Tim Finin. 1999. Yahoo! as an ontology: using Yahoo! categories to describe documents. In *Proceedings of the eighth international conference on Information and knowledge management, CIKM '99*, pages 180–187.
- Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. 2005. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explorations Newsletter*, 7(1):36–43, June.
- LSHTC3. 2012. *ECML/PKDD-2012 Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98.
- Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured Perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464.
- Arturo Montejo-Ráez and Luis Alfonso Ureña-López. 2006. Selection strategies for multi-label text categorization. In *Advances in Natural Language Processing*, pages 585–592. Springer.
- Kunal Punera and Joydeep Ghosh. 2008. Enhanced hierarchical classification via isotonic smoothing. In *Proceedings of the 17th international conference on World Wide Web, WWW '08*, pages 151–160.
- Xipeng Qiu, Wenjun Gao, and Xuanjing Huang. 2009. Hierarchical multi-label text categorization with global margin maximization. In *Proc. of the ACL-IJCNLP Short*, pages 165–168.
- Xipeng Qiu, Xuanjing Huang, Zhao Liu, and Jinlong Zhou. 2011. Hierarchical text classification with latent concepts. In *Proc. of ACL*, pages 598–602.
- Yutaka Sasaki and Davy Weissenbacher. 2012. TTI'S system for the LSHTC3 challenge. In *ECML/PKDD-2012 Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*.
- Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining multi-label data. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer.
- Xiao-Lin Wang, Hai Zhao, and Bao-Liang Lu. 2011. Enhance top-down method with meta-classification for very large-scale hierarchical classification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1089–1097.

# (Pre-)Annotation of Topic-Focus Articulation in Prague Czech-English Dependency Treebank

Jiří Mírovský, Kateřina Rysová, Magdaléna Rysová, Eva Hajičová

Charles University in Prague  
Faculty of Mathematics and Physics  
Institute of Formal and Applied Linguistics  
Czech Republic

{mirovsky|rysova|magdalena.rysova|hajicova}@ufal.mff.cuni.cz

## Abstract

The objective of the present contribution is to give a survey of the annotation of information structure in the Czech part of the Prague Czech-English Dependency Treebank. We report on this first step in the process of building a parallel annotation of information structure in this corpus, and elaborate on the automatic pre-annotation procedure for the Czech part. The results of the pre-annotation are evaluated, based on the comparison of the automatic and manual annotation.

## 1 Introduction

In the past three or four decades, topic-focus articulation (known also as sentence information structure) is a language phenomenon that has attracted an enormous interest in linguistics and has become a “hot” topic of linguistic studies. No wonder then, that these days several linguistic teams (e.g. at the University of Potsdam, University of Berlin, University of Stuttgart, Charles University in Prague) have attempted to include the annotation of information structure in the annotating schemes they propose. Among corpora that contain also annotation of information structure or such type of annotation is planned in them there are e.g. ANNIS database (Annotation of Information Structure, see Dipper et al., 2004), The English Switchboard Corpus (see Calhoun et al., 2005), the corpus DannPASS (Danish Phonetically Annotated Spontaneous Speech, see Paggio, 2006) and the Prague Dependency Treebank (for the information on PDT, see Hajič et al., 2006).

There are also several types of annotation guidelines and schemes for the different corpora, based on various linguistic theories dealing with information structure (e.g. Hajičová et al., 2000;

Nissim et al., 2004; Dipper et al., 2007; Donhauser, 2007; Cook and Bildhauer, 2011).

In our paper, we present the annotation of topic-focus articulation in the Czech part of the Prague Czech-English Dependency Treebank, based on the theory of topic-focus articulation as developed withing the Praguian Functional Generative Description. It is the first step in the process of building a parallel Czech-English corpus annotated with this type of linguistic information.<sup>1</sup>

### 1.1 Topic-Focus Articulation in Prague Treebanks

The first complex and consistent theoretically-based annotation of topic-focus articulation was already fully applied in the first Czech corpus from the Prague corpora family, the Prague Dependency Treebank (PDT; Hajič et al., 2006, updated in Bejček et al., 2012), and is available for the linguistic community. PDT is a large collection of Czech journalistic texts, (basically) manually annotated on several layers of language description (more than 3 thousand documents consisting of almost 50 thousand sentences are annotated on all the levels).

Detailed annotation guidelines that constitute the basis of the handling with the language material were developed (Mikulová et al., 2005) based on the theoretical assumptions of the Functional Generative Grammar (for the first formulations of this formal framework, see Sgall, 1967; Sgall et al., 1986). The annotation of the information structure in PDT is also based on this theory. The same linguistic approach was used in some other annotation schemes connected with the annotation of topic-focus articulation (e.g. Postolache, 2005).

<sup>1</sup> Given the available funds, our present goal is to annotate 5 thousand parallel sentences.

## 1.2 Aim of the Paper

Our effort is concentrated on annotating the topic-focus articulation (TFA) in a parallel corpus – the Prague Czech-English Dependency Treebank (PCEDT), to make possible contrastive studies of this phenomenon. As the first step, we annotate topic-focus articulation in the Czech part of the treebank. The annotation guidelines have been taken over from the PDT approach, i.e. they also follow the theory of Functional Generative Description.

In Section 2, we give an overview of the theoretical background of TFA, Section 3 introduces the Prague Czech-English Dependency Treebank (the data to be annotated). Section 4 describes in detail an automatic pre-annotation procedure that was applied on the data before they were annotated manually by a human annotator. The final step of this part of our research was the evaluation of effectiveness of the automatic pre-annotation, given in Section 5.

## 2 Theoretical Background for Corpus Annotation of Topic-Focus Articulation in PCEDT

The theoretical linguistic background for the creating of the whole corpus PCEDT is the Functional Generative Description (Sgall, 1967; Sgall et al., 1986). Topic-focus articulation in this theoretical framework was described especially by Sgall and Hajičová (summarized in Sgall et al., 1986, Hajičová et al., 1998). On the basis of this, the annotation guidelines for manual annotation of topic-focus articulation in the Prague Dependency Treebank (PDT) were established and are available in the annotation manual for the underlying structure of sentences in Mikulová et al. (2005). These guidelines are used also for the Czech part of the Prague Czech-English Dependency Treebank.

### 2.1 Topic-Focus Articulation in Functional Generative Description

The theory of topic-focus articulation within the framework of Functional Generative Description is based on the aboutness-principle: the topic is the part of a sentence that is spoken about, and, complementarily, the focus is the sentence part that declares something about the topic. From the cognitive point of view, topic may be characterized as the “given” part of the sentence and focus as the “new” one. However, this does not mean that the focus elements cannot be mentioned in

the previous language context at all but they have to bring some non-identifiable information or information in new relations.

Most sentences contain both parts – topic and focus. However, some sentences can be contextually independent (e.g. the first sentence of the text or its title) and they do not have to contain the topic part (these are topic-less sentences). On the contrary, the focus is an obligatory component of every sentence – it is the informatively more important part of the message than the topic.

The basic opposition established by the TFA theory and included in the annotation scheme is the opposition of contextual boundness: each element of the underlying structure of the sentence carries the feature “contextually bound” or “contextually non-bound”. In addition, the contextually bound elements in the topic can be either contrastive, or non-contrastive. Contrastive contextually bound sentence members differ from the non-contrastive ones in the presence of a contrastive stress and in their semantic content – they express contrast to some previous context (e.g. *at home – abroad*).

Non-contrastive contextually bound expressions are marked as 't', contrastive contextually bound expressions are marked as 'c' and contextually non-bound expressions are marked as 't'<sup>2</sup>.

The opposition between contextually bound and contextually non-bound elements serves then as a basis for the bi-partition of the sentence into its topic and focus; according to this hypothesis, an algorithm for topic-focus bi-partition was formulated, implemented and tested on the PDT data, with some rather encouraging results (see Hajičová et al., 2005).

In Czech (Czech is the language of Prague Dependency Treebank and also of one half of the Prague Czech-English Dependency Treebank), the word order position of predicative verb is often the natural boundary between the topic and focus part in the sentence – cf. Example (1).

(1) [Context: *Moje matka má ráda růže a tulipány.*] *Tulipány*<sub>contrastive\_topic</sub> *matka*<sub>topic</sub> *včera*<sub>topic</sub> *koupila*<sub>focus</sub> *na trhu*<sub>focus</sub>

Literally: [Context: *My mother likes roses and tulips.*] *The tulips*<sub>contrastive\_topic</sub> *the mother*<sub>topic</sub> *yesterday*<sub>topic</sub> *bought*<sub>focus</sub> *on the market*<sub>focus</sub>.

<sup>2</sup> The contextually non-bound elements do not have a contrastive and non-contrastive variant in the theory of FGP.

(= *The mother bought the tulips ON THE MARKET*<sup>3</sup> yesterday.)

Several operational tests have been proposed in literature that help to distinguish between topic and focus, the most relevant of them being the question test and the test of negation (for details see Sgall et al., 1986; Hajičová et al., 1998).

In short, the basis of the question test is to ask a question that fully represents the context for the tested sentence. The tested sentence has to be a relevant answer to the question. The sentence members present in both the question and answer are topic members. The elements present only in the answer are members of the focus.

The principle of the negation test is to find out the possible scope of negation in the negative counterpart to the given sentence. In principle, the sentence members that are in the scope of negation in the given context belong to the focus part of the sentence. Other members form the topic part. However, there is a possibility of negative topic, i.e. the topic of the sentence is negated and the focus stands out of the scope (for details see e.g. Sgall et al., 1973).

For detailed information on annotation guidelines of topic-focus articulation in the framework of Functional Generative Description, the online annotation manual is available (see <http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/index.html>).

### 3 Language Material – Prague Czech-English Dependency Treebank

The annotation effort described in this paper is performed on data from the Prague Czech-English Dependency Treebank (PCEDT, Hajič et al., 2012), a manually parsed parallel Czech-English corpus that contains over 1.2 million running words (50 thousand sentences in each of the two languages). The English part consists of texts from the Penn Treebank (Marcus et al., 1993) – articles from the Wall Street Journal. The Czech part contains human translations of the English sentences to Czech.

The annotation (on both language sides) is performed on four language layers: the “word” layer, the morphological layer, the analytical layer (i.e. the layer of surface syntax) and the tectogrammatical layer (i.e. the semantic layer of the deep syntax).

On the topmost (tectogrammatical) layer, individual sentences are organized in dependency

<sup>3</sup> The members that carry the centre of the intonation in the sentence are capitalized (in the translation).

tree structures, according to the style of the Prague Dependency Treebank (PDT). Autosemantic words and coordinating structures are captured in the trees, as well as the valency of verbs (each language has its own valency lexicon in PCEDT). Additionally, the surface sentence ellipsis is reconstructed in the deep sentence structure and also pronominal anaphoric relations are labeled in the texts. The topic-focus articulation is also to be annotated on this layer.

The parallel Czech-English data are aligned manually on the level of sentences and automatically on the level of tectogrammatical nodes.

More detailed information on PCEDT is available on the project website (<http://ufal.mff.cuni.cz/pcedt2.0/en/index.html>).

### 4 Automatic Pre-Annotation

For the annotation of topic-focus articulation in the Czech part of PCEDT, an automatic pre-annotation procedure was developed. The particular steps (rules) of the pre-annotation were mainly established on the basis of the completed annotation of contextual boundness in the Prague Dependency Treebank (i.e. on the basis of annotated Czech texts). The cross-language alignment of tectogrammatical nodes in PCEDT was also exploited (see the pre-annotation step 10 below), allowing for taking advantage of the existence of indefinite articles in English (not present in the Czech language).

Using information from the English side for the pre-annotation of topic-focus articulation in the Czech part is possible, as the topic-focus articulation of the given sentence in the given context should be identical regardless on the language<sup>4</sup>. The surface word order may vary in Czech in comparison with English (cf. the different word order in Example (1) in the two languages) but the topic-focus articulation of the sentence should be the same in both the languages. This theoretical assumption, as well as the quality of the English->Czech translation (from the point of view of topic-focus articulation), can be tested on real corpus data once the annotation on both language sides of PCEDT is finished.

<sup>4</sup> In fact, the topic-focus articulation of the given sentence is the same regardless on the language. However, we operate with a parallel corpus – the English part contains original texts and the Czech one their translations. It is possible that the Czech translations could be inaccurate in some cases – especially regarding the topic-focus articulation. Therefore, the value of contextual boundness could differ in both parts of parallel corpus in a few cases.

So far, the automatic procedure was used for pre-annotation of a sample of the PCEDT Czech part and this pre-annotated sample was subsequently manually annotated by a human annotator. The annotator checked the correctness of the pre-annotation and annotated the rest of the nodes (nodes that had not been pre-annotated). Afterwards, it was evaluated how many changes of the automatic pre-annotation of topic-focus articulation the human annotator had to carry out, i.e. how many mistakes the automatic pre-annotation had made in the data.

It should be noted that the goal of the automatic pre-annotation was to help the human annotators with simple decisions, not to classify every sentence member as contextually bound ('t') or non-bound ('f') element. Our intention was to apply only reliable rules and leave too complex decisions (often depending on the meaning of the text) on the human annotator. We wanted to avoid introducing too many errors in the pre-annotation, as human annotators might be prone to overlooking errors in already annotated nodes and concentrate only (or at least better) on the so far unannotated nodes. For the selection of the pre-annotation steps, we estimated their expected error rates (where possible) based on measurements on the topic-focus annotation in PDT (see the expected error rates of the individual pre-annotation steps below in 4.1). For using a rule, we set the maximum number of expected errors to 10 %.

#### 4.1 Steps of the Pre-Annotation

The following steps have been performed during the automatic pre-annotation. For each step (where possible), we give an estimate of the pre-annotation error (expected error rate, EER), based on the measurement of the phenomenon in the data of Prague Dependency Treebank. The steps have been applied in the presented order. Step 10 takes advantage of the cross-language alignment of words in PCEDT.

1. **Nodes generated** on the tectogrammatical layer **without a counterpart on the analytical layer** (i.e. newly added, but not copied nodes in the tectogrammatical representation) and that do not have functor=RHEM (rhematizer), nor t\_lemma=#Forn (part of a phrase in a foreign language), get automatically assigned tfa='t', i.e. contextually bound, (EER: 0). For an example, see Figure 1.<sup>5</sup>

<sup>5</sup> Sentence members (nodes) that are really expressed in the surface sentence structure (that appear on both the analytical

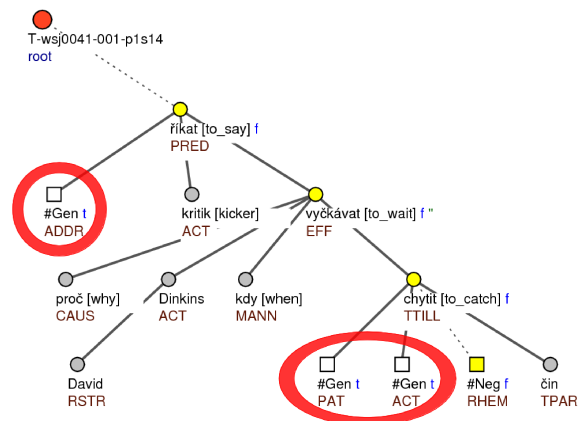


Figure 1: Example of a sentence tree structure in the Czech part of PCEDT: circled nodes represent the automatically pre-annotated sentence members marked as 't' (contextually bound)

Figure 1 represents the following Czech sentence – Example (2) from PCEDT:

(2) „Proč David Dinkins,“ říká kritik, „vždycky vyčkává, dokud není chyten při činu?“  
 “David Dinkins,” says the kicker,  
 “Why does he always wait until he's caught?”

In the surface (analytical) structure of the given sentence with the Czech verb *řikat* (to say), the Addressee is not present explicitly although this verb has the Addressee (apart from the Effect, the Actor and the non-obligatory Patient) in its valency frame (*someone*.obligatory\_Actor says *something*.obligatory\_Effect to *someone*.obligatory\_Addressee about *something/somebody*.non-obligatory\_Patient). So the Addressee is present only in the deep (tectogrammatical) sentence structure (in Figure 1, it is captured as a small square with the symbol of Addressee ADDR). The sentence members that appear only implicitly in the sentence (as the Addressee in this case) are not supposed to carry some new, important information (because their presence in the /surface part of the/ sentence is not necessary) and therefore they are automatically pre-annotated as contextually bound (fur-

and the tectogrammatical layer) are displayed as small circles in the figure. Members that are present only in the deep sentence structure (on the tectogrammatical layer) and do not appear in the surface sentence structure (i.e. not on the analytical layer) are displayed as small squares.

White colour represents contextually bound sentence members (they are also depicted with 't' next to the lemma); yellow colour (light grey in b/w) represents contextually non-bound sentence members (they are depicted with 'f'). The grey members do not have any value of contextual boundness yet (they were not automatically pre-annotated and they will be manually annotated by a human annotator).



ther examples are the sentence members Patient PAT and Actor ACT by the Czech verb *chytiť* – *to catch*: *somebody*<sub>obligatory\_Actor</sub> *catches some-one*<sub>obligatory\_Patient</sub>, see Figure 1).

2. **Nodes generated at the tectogrammatical layer that are members of coordination/apposition** and have an analytical counterpart (they are copied nodes; it also means that it is not e.g. #Forn), get assigned  $tfa='t'$ , i.e. contextually bound, (EER: 0), see Example (3) from PCEDT.

(3) „Nyní,“ říká Joseph Napolitan, průkopník politické televize, „je cílem jít do útoku jako první, poslední a [jít]<sub>t</sub> vždycky.“

“Now,“ says Joseph Napolitan, a pioneer in political television, “the idea is to attack first and [to attack]<sub>t</sub> always.”

This pre-annotation step concerns also other cases of sentence members that are not present in the surface (analytical) structure but appear in the deep (tectogrammatical) layer. These nodes are not newly added to the structure, e.g. because of the valency verb frame, but they appeared in some previous structures and they are omitted in the surface structure (and copied to the deep structure) because the reader can understand them easily from the previous context as in the phrases from Example (3): *to attack first* and *(to attack) always*. Since these members (present only implicitly in the sentence) are obviously deducible from the context, they are considered as contextually bound and therefore they are pre-annotated as such.

3. **Nodes where a grammatical, textual or segment coreference starts**, get  $tfa='t'$ , i.e. contextually bound, (EER: 1:100), see Example (4) from PCEDT.

(4) A *Dinkins* podle *svých<sub>t</sub>* slov nevěděl, že *muž, kterého<sub>t</sub>* platili v rámci kampaně za přesvědčování voličů k účasti, byl odsouzen za únos.

And, says Mr. *Dinkins*, *he<sub>t</sub>* didn't know the *man his<sub>t</sub>* campaign paid for a get-out-the-vote effort had been convicted of kidnapping.

This step of the automatic pre-annotation takes advantage of the finished annotation of coreference in the PCEDT texts. Sentence elements that are anaphors<sup>6</sup> of a coreference relation are sup-

<sup>6</sup> A reference to an entity or event that has already been mentioned in the preceding text; the two mentions –

posed to be contextually bound and therefore they are automatically assigned the value 't'.

There are two coreference relations in Example (4): 1. *Dinkins* – *svých (he)*; 2. *muž (man) – kterého (his)*. The members that refer to some previous sentence members (*svých* and *kterého* in this case) are automatically pre-annotated as contextually bound.

In another example from PCEDT, depicted in Figure 2, starting nodes (anaphors) of grammatical coreference (three intra-sentential more or less vertical arrows) and textual coreference (two horizontal arrows going from the second tree to the first one) are pre-annotated as contextually bound.

4. Nodes with **functor=PRED** that are **not newly generated** and whose  $t\_lemma$  does not appear in the previous sentence, get  $tfa='f'$ , i.e. contextually non-bound, (EER: 1:40), see Example (5) from PCEDT.

(5) „Pamatujete si na Pinocchia?“ říká<sub>f</sub> ženský hlas.

“Remember Pinocchio?“ says<sub>f</sub> a female voice.

The data of previously annotated Prague Dependency Treebank demonstrated that most Predicates (in corpus marked as PRED) are contextually non-bound – therefore, they are pre-annotated as 'f'.

5. **Newly generated nodes with functor=PRED** get  $tfa='t'$ , i.e. contextually bound, (EER: 1:100), see Example (6) from PCEDT.

In contrast to the step 4), Predicates that are not present in the surface sentence structure are pre-annotated as contextually bound, cf. step 3).

(6) Na obrazovce vidíme dvě zkreslené rozmazané fotografie, **pravděpodobně<sub>MOD.f</sub>** [vidíme]<sub>t</sub> fotografie dvou politiků.

The screen shows two distorted, unrecognizable photos, **presumably<sub>MOD.f</sub>** [shows]<sub>t</sub> [photos] of two politicians.

6. **Other verbal nodes** (gram/sempos=v) with **functor** from the set {ADDR, AIM, CAUS, ACMP, MANN, PAT, EFF, AUTH, BEN, COMPL, EXT, ORIG, RESL, TFHL, TSIN} get  $tfa='f'$ , i.e. contextually non-bound, (EER: 1:10), see Example (7) from PCEDT.

anaphor (the latter in the text) and antecedent (the former) are connected by a coreference relation.



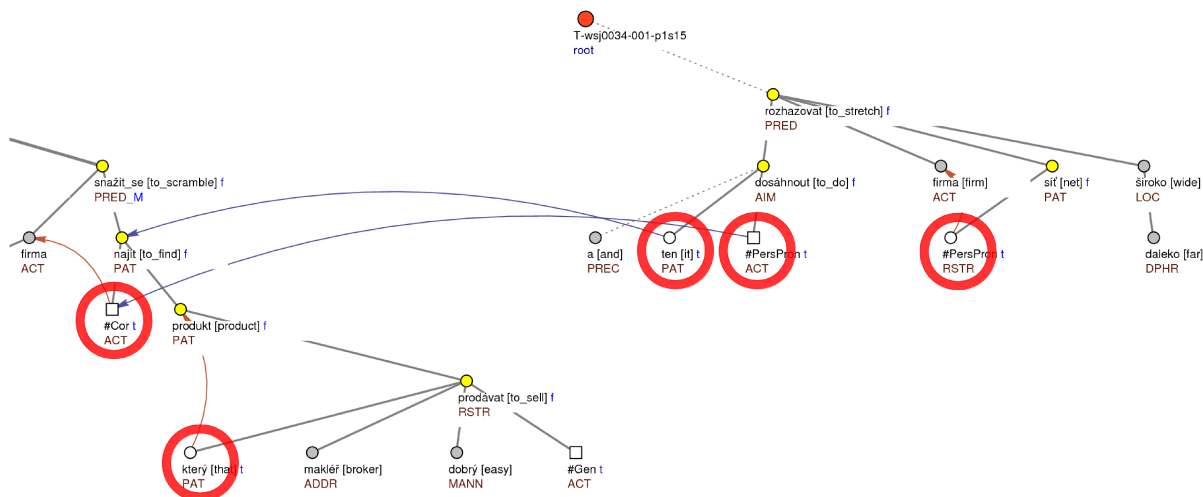


Figure 2: Two trees representing two sentences; the indexes in the sentences and the arrows in the trees denote coreference chains; starting nodes of the coreference links are marked by large circles: *Soukromí investoři se s léty od burzy odvracejí a investiční firmy[2] se snaží [firmy][2] najít[1] produkty[4], které[4] by se makléřům dobře prodávaly. A aby toho[1] [firmy][2] dosáhly, rozhazují firmy[3] své[3] síť široko daleko. (In original: As individual investors have turned away from the stock market over the years, securities firms[2] have scrambled to [firms][2] find[1] new products[4] that[4] brokers find easy to sell. And the firms[3] are stretching their[3] nets far and wide to [firms][2] do it[1].)*

Note that there is no link between *firm* in the first tree and *firm* in the second tree, as only pronominal coreference is annotated in the data. Otherwise, chains [2] and [3] would be one coreference chain.

The example has been cropped to fit the page (the left part of the first tree).

The data of the Prague Dependency Treebank also demonstrated that most sentence members expressed as dependent clauses (i.e. containing a finite verb) and having the semantic role of Addressee, Aim, Cause, Accompaniment, Patient, Effect, Author, Benefactor, Complement, Extent, Origo, Result or Temporal modifications (expressing *for how long* or *since when*) are contextually non-bound – therefore, they are pre-annotated as non-bound also in data of the Prague Czech-English Dependency Treebank.

(7) „Porovnejte tyto dva kandidáty na starostu.“.Effect<sub>f</sub> říká hlasatel.  
 “Compare two candidates for mayor.“.Effect<sub>f</sub> says the announcer.

7. Nodes with **functor** from the set {PARTL, DENOM, MOD, EXT} get tfa='f', i.e. contextually non-bound, (EER: 1:10), see again Example (6) above from PCEDT.

The data of the Prague Dependency Treebank further demonstrated that most sentence members assigned the semantic role of independent interjectional clause (marked as PARTL), independent non-parenthetical nominal clause (DENOM), atomic expression with a modal meaning (MOD) or adjunct expressing extent (EXT) are contextually non-bound and therefore they are pre-annotated as such.

In the Example (6), the sentence member *pravděpodobně (presumably)* is in the role of an atomic expression with a modal meaning (MOD) and therefore it will be automatically assigned the value 'f'.

8. Nodes with **functor=RHEM** (i.e. they have a function of a rhematizer) that are not in the first position in the sentence, get tfa='f', i.e. contextually non-bound, (EER: 1:10), see Example (8) from PCEDT.

(8) Letošek je rokem, kdy se negativní reklama, po léta přítomná ve většině politických kampaní jen<sub>f</sub> druhotně, stala hlavní událostí.  
 This is the year the negative ad, for years [only]<sub>f</sub> a secondary presence in most political campaigns, became the main event.

The rhematizers (as e.g. English particles *only, for example, also, especially, principally*) mostly precede a focus element and in the theory of TFA, they are also considered contextually non-bound. However, also contrastive contextually bound expressions can follow the rhematizers – typically at the beginning of the sentence (and in this case, also the rhematizers are contextually bound). Therefore, only such rhematizers are pre-annotated as contextually non-bound that are not placed in the initial position in the sentence.

9. Nodes with **t\_lemma=tady (here)** get  $tfa='t'$ , i.e. contextually bound, (EER: 1:10), see Example (9) from PCEDT.

Some lemmas (especially with a deictic function like *here*) appear as contextually bound in most cases (but not in all – see e.g. *What happens here<sub>t</sub> and now?*), which observation is also made use of in the automatic pre-annotation.

(9) *Ředitelka Wardová se rozhodla zbavit se „balastu“ v učitelském sboru a obnovit bezpečnost a také tu<sub>t</sub> byly další nové faktory, které pracovaly v její prospěch.*

*Mrs. Ward resolved to clean out “deadwood” in the school’s faculty and restore safety, and she also had some new factors [here]<sub>t</sub> working in her behalf.*

10. Nodes that are **Czech counterparts of English nodes** that in the English sentence are placed after their governing verb on the surface and that are **preceded by an indefinite article**, get  $tfa='f'$ , i.e. contextually non-bound, (EER: unknown), see Example (10) from PCEDT.

(10) *The war over federal judicial salaries takes a victim.* ↓  
*Válka o platy federálních soudců si žádá svou první oběť<sub>f</sub>.*

In Example (10), the sentence member *victim* is modified by the indefinite article *a* in the English variant of the sentence, which leads to the assumption that this member is contextually non-bound. Since the value of the same sentence member should be identical both in English and in Czech variant of the sentence, also the Czech member *oběť* (that is the counterpart of the *victim*) is supposed to be contextually non-bound.

The following steps of the automatic pre-annotation are performed after the previous steps have been applied on all nodes of the given tree:

11. **Daughters of a verb that has  $tfa='f'$**  and that is not on the first or second position (in its clause), if they appear after the governing verb on the surface, get  $tfa='f'$ , i.e. contextually non-bound, (EER: unknown), see Example (11) from PCEDT.

(11) *Na konci druhé světové války se Německo vzdalo<sub>f</sub> dříve než Japonsko<sub>f</sub>...*

*At the end of World War II, Germany surrendered<sub>f</sub> before Japan<sub>f</sub>...*

This step of the pre-annotation makes use of the fact that in Czech, the surface word order often is used to express the topic-focus articulation. Under the condition that the contextually non-bound predicative verb is placed further to the right than on the second position in the sentence and that the sentence has a non-marked word order<sup>7</sup> (i.e. emotionally neutral), it is possible to assume that the sentence members following the predicative verb are contextually non-bound.

12. Nodes with **functor=RSTR** that are **daughters of a node with  $tfa='f'$** , get  $tfa='f'$ , i.e. contextually non-bound, (EER: 1:30).

(12) *Zasedání společného výboru sněmovny a senátu se koná v případě, že sněmovna a senát schválí zákon v odlišné<sub>f</sub> podobě.*

*The Senate-House conference committee is used when a bill is passed by the House and Senate in different<sub>f</sub> forms.*

The final step of the automatic pre-annotation is based on the fact that the adnominal adjuncts modifying its governing noun (in the annotated corpus marked as RSTR) often have a very high degree of communicative dynamism because their primary function is to specify something. Therefore, they are pre-annotated as contextually bound (if they modify a non-bound element at the same time).

## 5 Evaluation of the Automatic Pre-Annotation

At the time of submitting the final version of the paper, more than one thousand automatically pre-annotated sentences have also been manually annotated by a human annotator<sup>8</sup> and could be used for evaluation of the pre-annotation.

In 59 documents (1,145 sentences, 22,436 nodes on the tectogrammatical layer), 7,864 nodes out of 19,105  $tfa$ -relevant nodes have been automatically pre-annotated (i.e. 41.1 %).

Table 1 gives an overview of how many times the individual pre-annotation steps have been applied. Based on the estimates presented in Sec-

<sup>7</sup> The human annotator decides whether the word order is marked or non-marked (it is not possible to check it automatically in our procedure of pre-annotation).

<sup>8</sup> There were actually two annotators, working on different parts of the data. For simplicity, we refer to them as 'a human annotator'. Only during a training phase (performed on a few documents), the two annotators worked on the same data and their discrepancies were subsequently checked by an arbiter and discussed.

tion 4.1 (for the two unknown estimates in steps 10 and 11 we used EER: 1:10), we can calculate the expected number of errors in the pre-annotation as (about) 340 errors.

step	short description	count
1	generated, no a-counterpart	1,988
2	generated, member of coord/app	127
3	anaphor of a coreference	742
4	PRED, not generated	1,189
5	PRED, generated	0
6	other verbal nodes (set of func.)	825
7	set of functors	435
8	RHEM (not first in sentence)	366
9	t_lemma=tady (here)	8
10	indefinite article in English	779
11	subseq. daughter of a verb in focus	237
12	RSTR daughters of a node in focus	1,168

Table 1: Usage of the individual pre-annotation steps

In the manual annotation, the annotator changed the pre-annotated value in 294 cases (i.e. 3.7 % of pre-annotated nodes). Table 2 shows details on the manually performed changes.

	pre-annotated value	
	't'	'f'
changed to 'c'	11	26
changed to 't'	-	244
changed to 'f'	13	-
no change	2,841	4,729

Table 2: The distribution of changes of automatically pre-annotated TFA-values manually made by human annotators

The numbers show that the automatic pre-annotation is more successful in marking contextually bound sentence members, as only 0.8 % of nodes pre-annotated as 't' and 5.4 % of nodes pre-annotated as 'f' were manually changed to another value.

	PDT 2.0	sample of PCEDT
contr. contextually bound ('c')	5.4 %	5.7 %
non-contr. contextually bound ('t')	31.3 %	33.6 %
contextually non-bound ('f')	63.3 %	60.7 %

Table 3: The percentage distribution of manually annotated TFA-values in PDT (training data) and so far annotated sample of the Czech part of PCEDT

The inability of the pre-annotation procedure to set the 'c' value (contrastive contextually bound) does not harm the results much, as only 37 (0.5 %) pre-annotated nodes were manually changed to this value, and the overall ratio of contrastive contextually bound nodes among all (manually) annotated nodes both in PDT and PCEDT is less than 6 % (see Table 3).

The main limitations of the pre-annotation are in its coverage (more than half of the nodes are not pre-annotated) and in its natural inability to take the meaning of the text into account (and thus being unable to better distinguish between 't' and 'f' values).

From another point of view, the results suggest that the expected error rates (estimated on PDT) are accurate and that the automatic pre-annotation is sufficiently reliable and serves as a substantial help to the annotators.<sup>9</sup>

## 6 Conclusion

The paper presented the first part of the project of parallel annotation of topic-focus articulation in the Prague Czech-English Dependency Treebank (PCEDT). We described the annotation principles and schemes, and elaborated on 12 automatic steps of the pre-annotation procedure for the Czech part of the treebank. The pre-annotation is able to mark over 40 % of the whole text (the rest is supposed to be annotated by human annotators). It can distinguish between contextually bound and non-bound sentence elements with the average success rate over 96 %, as shown by the evaluation on manually annotated texts.

## Acknowledgment

We gratefully acknowledge support from the Grant Agency of the Czech Republic (grants P406/12/0658 and P406/2010/0875). This work has been using language resources developed and/or stored and/or distributed by the LINDAT-Clarín project of the Ministry of Education of the Czech Republic (project LM2010013).

## References

- E. Bejček, J. Panevová, J. Popelka, P. Straňák, M. Ševčíková, J. Štěpánek, Z. Žabokrtský. 2012. Prague Dependency Treebank 2.5 – a revisited version of PDT 2.0. In: *Proceedings of the 24th Inter-*

<sup>9</sup> Of course, it is a matter of discussion (and testing), how much effort of the human annotator such a pre-annotation saves and how to set the reliability limit for the rule selection.

- national Conference on Computational Linguistics (Coling 2012)*, Mumbai, India, pp. 231–246.
- S. Calhoun, M. Nissim, M. Steedman, J. Brenier. 2005. A Framework for Annotating Information Structure in Discourse. In: *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 45–52. URL <http://aclweb.org/anthology/W/W05/W05-0307>.
- P. Cook, F. Bildhauer. 2011. Annotating information structure. The case of "topic". In: S. Dipper & H. Zinsmeister (eds.), *Beyond Semantics. Corpus based Investigations of Pragmatic and Discourse Phenomena*, Ruhr Universität Bochum, Bochumer Linguistische Arbeitsberichte, pp. 45–56. URL [http://www.linguistics.ruhr-uni-bochum.de/bla/beyondsem2011/cook\\_final.pdf](http://www.linguistics.ruhr-uni-bochum.de/bla/beyondsem2011/cook_final.pdf).
- S. Dipper, M. Götze, S. Skopeteas (eds.). 2007. *Information Structure in Cross-Linguistic Corpora: Annotation Guidelines for Phonology, Morphology, Syntax, Semantics and Information Structure*, vol. 7 of Interdisciplinary Studies on Information Structure. Potsdam, Germany: Universitätsverlag Potsdam. URL <http://www.sfb632.uni-potsdam.de/publications/isis07.pdf>.
- S. Dipper, M. Götze, M. Stede, T. Wegst. 2004. AN-NIS: A Linguistic Database for Exploring Information Structure. In Ishihara, S., Schmitz, M., Schwarz, A. (Eds.), *Working Papers of the SFB632, Interdisciplinary Studies on Information Structure (ISIS) 1*, pp. 245–279. Potsdam: University publishing house Potsdam.
- K. Donhauser. 2007. Zur informationsstrukturellen Annotation sprachhistorischer Texten. *Sprache und Informationsverarbeitung 31*, pp. 39–45. URL [http://www.sfb632.uni-potsdam.de/publications/B4/donhauser\\_2007.pdf](http://www.sfb632.uni-potsdam.de/publications/B4/donhauser_2007.pdf).
- J. Hajič, J. Panevová, E. Hajičová, P. Sgall, P. Pajas, J. Štěpánek, J. Havelka, M. Mikulová, Z. Žabokrtský, M. Ševčíková-Razímová. 2006. *Prague Dependency Treebank 2.0*. Software prototype, Linguistic Data Consortium, Philadelphia, PA, USA, ISBN 1-58563-370-4, <http://www ldc.u-penn.edu>, Jul 2006.
- J. Hajič, E. Hajičová, J. Panevová, P. Sgall, O. Bojar, S. Cínková, E. Fučíková, M. Mikulová, P. Pajas, J. Popelka, J. Semecký, J. Šindlerová, J. Štěpánek, J. Toman, Z. Urešová, Z. Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, European Language Resources Association, Istanbul, Turkey, ISBN 978-2-9517408-7-7, pp. 3153–3160.
- E. Hajičová, J. Havelka, K. Veselá. 2005. Corpus Evidence of Contextual Boundness and Focus. In: *Proceedings of the Corpus Linguistics Conference Series*, University of Birmingham, Birmingham, UK, ISSN 1747-9398.
- E. Hajičová, J. Panevová, P. Sgall. 2000. A Manual for Tectogrammatical Tagging of the Prague Dependency Treebank. *Technical report tr-2000-09*, ÚFAL/CKL. URL [http://ufal.mff.cuni.cz/pdt/Corpora/PDT\\_1.0/Doc/tmanual/tmanen.pdf](http://ufal.mff.cuni.cz/pdt/Corpora/PDT_1.0/Doc/tmanual/tmanen.pdf). In cooperation with A. Böhmová, M. Ceplová and V. Řezníčková. Translated by Z. Kirschner, E. Hajičová and P. Sgall.
- E. Hajičová, B. H. Partee, P. Sgall. 1998. *Topic-focus articulation, tripartite structures, and semantic content*. Dordrecht, Boston: Kluwer Academic Publishers.
- M. P. Marcus, B. Santorini, M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), pp. 313–330.
- M. Mikulová et al. 2005. *Annotation on the tectogrammatical layer in the Prague Dependency Treebank. The Annotation Guidelines*. Prague: ÚFAL MFF. Available at: <http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/index.html>.
- M. Nissim, S. Dingare, J. Carletta, M. Steedman. 2004. An annotation scheme for information status in dialogue. In: *Proceedings of the 4th Conference on Language Resources and Evaluation*. Lisbon, Portugal. URL <http://www.lrec-conf.org/proceedings/lrec2004/pdf/638.pdf>.
- P. Paggio. Annotating Information Structure in a Corpus of Spoken Danish. 2006. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pp. 1606–1609, Genoa, Italy.
- O. Postolache. 2005. Learning Information Structure in The Prague Treebank. In: *Proceedings of the ACL Student Research Workshop*, pp. 115–120, Ann Arbor, Michigan, June 2005.
- P. Sgall. 1967. *Generative description of language and the Czech Declension (in Czech)*. Prague: Academia.
- P. Sgall, E. Hajičová, E. Benešová. 1973. *Topic, focus and generative semantics (Vol. 1)*. Kronberg Taunus: Scriptor Verlag.
- P. Sgall, E. Hajičová, J. Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Springer.

# Animacy Acquisition Using Morphological Case

**Riyaz Ahmad Bhat**

LTRC, IIIT-Hyderabad, India

riyaz.bhat@research.iiit.ac.in

**Dipti Misra Sharma**

LTRC, IIIT-Hyderabad, India

dipti@iiit.ac.in

## Abstract

Animacy is an inherent property of entities that nominals refer to in the physical world. This semantic property of a nominal has received much attention in both linguistics and computational linguistics. In this paper, we present a robust unsupervised technique to infer the animacy of nominals in languages with rich morphological case. The intuition behind our method is that the control/agency of a noun depicted by case marking can approximate its animacy. A higher control over an action implies higher animacy. Our experiments on Hindi show promising results with  $F_\beta$  and *Purity* scores of 89 and 86 respectively.

## 1 Introduction

Animacy can either be defined as a biological property or a grammatical category of nouns. In a strictly biological sense, living entities are animate, while all non living entities are inanimate. However, in its linguistic sense, the term is synonymous with a referent's ability to act or instigate events volitionally (Kittilä et al., 2011). Although seemingly different, linguistic animacy can be implied from biological animacy. In linguistics, the manifestation of animacy and its relevance to linguistic phenomena have been studied quite extensively. Animacy has been shown, cross linguistically, to control a number of linguistic phenomena. Case marking, argument realization, topicality or discourse salience are some phenomena highly correlated with the property of animacy (Aissen, 2003; Bresnan et al., 2007; De Swart et al., 2008; Branigan et al., 2008). In linguistic theory, how-

ever, animacy is not seen as a dichotomous variable, rather a range capturing finer distinctions of linguistic relevance. Animacy hierarchy proposed in Silverstein's influential article on "animacy hierarchy" (Silverstein, 1986) ranks nominals on a scale of the following gradience: *1st pers* > *2nd pers* > *3rd anim* > *3rd inanim*. Several such hierarchies of animacy have been proposed following (Silverstein, 1986). One basic scale taken from (Aissen, 2003) makes a three-way distinction as *humans* > *animates* > *inanimates*. These hierarchies can be said to be based on the likelihood of a referent of a nominal to act as an agent in an event (Kittilä et al., 2011). Thus higher a nominal on these hierarchies higher the degree of agency/control it has over an action. In morphologically rich languages, the degree of agency/control is expressed by case marking. Case markers capture the degree of control a nominal has in a given context (Hopper and Thompson, 1980; Butt, 2006). They rank nominals on the continuum of control as shown in (1)<sup>1</sup>. Nominals marked with Ergative case have highest control and the ones marked with Locative have lowest.

$$\text{Erg} > \text{Gen} > \text{Inst} > \text{Dat} > \text{Acc} > \text{Loc} \quad (1)$$

In this work, we demonstrate that the correlation between the aforementioned linguistic phenomena is highly systematic, therefore can be exploited to predict the animacy of nominals. In order to utilize the correlation between these phenomena for animacy prediction, we choose to use an unsupervised learning method. Since, using a supervised learning technique is not always feasible. The resources required to train supervised algorithms are expensive to create and unlikely to

<sup>1</sup>Ergative, Genitive, Instrumental, Dative, Accusative and Locative in the given order.

exist for the majority of languages. We show that an unsupervised learning method can achieve results comparable to supervised learning in our setting (see Section 5). Further, based on our case study of Hindi, we propose that given the morphological case corresponding to Scale (1), animacy can be predicted with high precision. Thus, given the morphological case our approach should be portable to any language. In the context of Indian languages, in particular, our approach should be easily extendable. In many Indo-Aryan languages<sup>2</sup>, the grammatical cases listed on Scale (1) are, in fact, morphologically realized (Masica, 1993, p. 230) (Butt and Ahmed, 2011).

In what follows, we first present the related work on animacy acquisition in Section 2. In Section 3, we will describe our approach for acquiring animacy in Hindi using case markers listed in (2). Section 3.1 describes the data used in our experiments, followed by discussion on feature extraction and normalization. In Section 4, we discuss the extraction of data sets from Hindi Wordnet for the evaluation of results of our experiments. In Section 5, we describe the results with thorough error analysis and conclude the paper with some future directions in Section 6.

## 2 Related Work

In NLP, the role of animacy has been recently realized. It provides important information, to mention a few, for anaphora resolution (Evans and Orasan, 2000), argument disambiguation (Dell’Orletta et al., 2005), syntactic parsing (Øvrelid and Nivre, 2007), (Bharati et al., 2008) and verb classification (Merlo and Stevenson, 2001). Lexical resources like wordnet usually feature animacy of nominals of a given language (Fellbaum, 2010; Narayan et al., 2002). However, using wordnet, as a source for animacy, is not straightforward. It has its own challenges (Orsan and Evans, 2001; Orsan and Evans, 2007). Also, it’s only a few privileged languages that have such lexical resources available. Due to the unavailability of such resources that could provide animacy information, there have been some notable efforts in the last few years to automatically acquire animacy. The important and worth mentioning works in this direction are (Øvrelid, 2006) and (Øvrelid, 2009). The works focus on Swedish and Norwegian common nouns using dis-

---

<sup>2</sup>Indo-Aryan is a major language family in India.

tributional patterns regarding their general syntactic and morphological properties. Other works in the direction are (Bowman and Chopra, 2012) for English and (Baker and Brew, 2010) for English and Japanese. All these works use supervised learning methods on a manually labeled data set. These works use highly rich linguistic features (e.g., grammatical relations) extracted using syntactic parsers and anaphora resolution systems. The major drawback of these approaches is that they can not be extended to resource poor languages because these languages can not satisfy the prerequisites of these approaches. Not only the availability of manually annotated training data, but also the features used restrict their portability to resource poor languages. Our approach, on the other hand, is based on unsupervised learning from raw corpus using a small set of case markers. Therefore, it can be extended to any language with morphologically realized grammatical case listed on Scale (1).

## 3 Our Approach

As noted by Comrie (1989, p. 62), a nominal can have varying degrees of control in varying contexts irrespective of its animacy. The noun phrase *the man*, for example, is always high in animacy, but it may vary in degree of control. It has high control in *the man deliberately hit me* and minimal control in *I hit the man*. In morphologically rich languages, case markers capture the varying control a nominal has in different contexts. In Hindi, for example, a nominal, in contexts of high control, occurs with a case marker listed high on hierarchy (1) (e.g., ergative), while in contexts of low control is marked with a case marker low on (1) (e.g., locative). Because of the varying degrees of control a nominal can have across contexts, approximating animacy from control would be misleading. Therefore, we generalize the animacy of a nominal from its overall distributions in the corpora. Now the question is, how to generalize the animacy from the mixed behavior that a nominal displays in a corpora? The linguistic notion of markedness addresses this problem. An unmarked observation, in linguistics, means that it is more frequent, natural, and predictable than a marked observation (Croft, 2002). Although, a given nominal can have varying degrees of control in different contexts irrespective of its animacy, its unmarked behavior should correlate well with

its literal animacy, i.e., animates should more frequently be used in contexts of high control while in-animates should be used in contexts of low control. A high degree of animacy necessarily implies high degree of control. So the prototypical use of animates is in the contexts of high control and of inanimates in the contexts of low control. As the discussion suggests, animates should occur more frequently with the case markers towards the left of the Scale (1), while inanimates should occur more frequently with the ones towards the right of the Scale. Thus, animates should have a left-skewed distribution on Scale (1), while inanimates should have a right-skewed distribution.

In this work, we have exploited the systematic correlations between the linguistic phenomena, as discussed, to approximate animacy of Hindi nominals. Our methodology relies on the distributional patterns of a nominal with case markers capturing its degree of control. Distributions of each nominal are extracted from a large corpus of Hindi and then they are clustered using fuzzy *cmeans* algorithm. Next, we discuss our choice of clustering, feature extraction and normalization.

### 3.1 Feature Extraction and Normalization

In order to infer animacy of a nominal, we extracted its distributions with the case markers corresponding to (1) except genitives<sup>3</sup>. Case markers of Hindi corresponding to (1) are listed in (2) (Mohan, 1990, p. 72).

$$\text{ne} > \text{kaa} > \text{se} > \text{ko} > \text{ko} > \{\text{mem, par, tak, se, ko}\} \quad (2)$$

Since *ko* and *se* are ambiguous, as shown in (2), we approximated them to the prototypical cases they are usually used for. *ko* is approximated to dative while *se* is approximated to instrumental case. The ambiguity in these case makers, however, has a profound impact on our results as discussed in Section 5. A mixed-domain corpora of 87 million words is used to ensure enough case marked instances of a nominal. The extraction of distributional counts is simple and straightforward in Hindi. Words immediately preceding case markers are considered as nouns since case markers almost always lie adjacent to the nominals they mark, however, occasionally they are separated by emphatic particles like *hi* ‘only’. In such cases particles are removed to extract the distribution by

<sup>3</sup>Genitives are highly ambiguous in Hindi and hardly discriminate animates from in-animates.

using a list of stop words. Since, Hindi nouns decline for number, gender and case, we use Hindi morph-analyzer, built in-house, to generate lemmas of inflected word forms so that their distributions can be accumulated under their corresponding lemmas. Further, the distributional counts of each nominal are scaled to unity so as to guard against the bias of word frequencies in our clustering experiments. Consider a distribution of two nominals *A* and *B* with case markers *X* and *Y*. Say *A* occurs 900 times with *X* and 100 times with *Y* and *B* occurs 18 times with *X* and 2 times with *Y*. Although, these nominals seem to have different distributions, apart from being similarly skewed, both of them have similar relative frequency of occurrence with *X* and *Y*. We aim, therefore, to normalize the distributional counts of a nominal with the case markers it occurs with. The distributional counts are normalized to unity by the frequency of a given nominal in the corpora, as shown in (3). This ensures that only the nominals of similar relative frequency distributions are clustered together. Beside, normalizing the distributions, we set a frequency threshold, for a nominal to be included for clustering to  $> 10$ , which ensures its enough instances to unravel its unmarked or prototypical behavior.

$$x' = \frac{x_i}{\sum_{i=1}^k x_i} \quad (3)$$

$x'$  is the normalized dimensions in a feature vector of a nominal  $x$ .  $k$  is the number of coordinates and  $x_i$  is the  $i^{\text{th}}$  coordinate of  $x$ .

### 3.2 Soft Clustering

Animacy is an inherent and a non varying property of entities that nominals refer to. However, due to lexical ambiguity animacy of a nominal can vary as the context varies. In Hindi, the ambiguity can be attributed to the following:

- **Personal Names:** In Hindi, common nouns are frequently used as person names or as a component of them. For example, noun ‘baadal’ meaning ‘cloud(s)’ can also be used as a ‘person name’; similarly ‘vijay’ can either mean ‘victory’ or can be a ‘person name’.
- **Metonymies:** Metonymies or complex types (logical polysemy) like institute names, country names etc, can refer to a building,

a geographical place or a group of individuals depending on the context of use. These words are not ambiguous per se but show different aspects of their semantics in different contexts (logically polysemous). For example, *India* can either refer to a geographical place or its inhabitants.

These ambiguities imply that some nominals can belong to both animate and inanimate classes. In order to address this problem of mixed membership, we used soft clustering approach in this work. In comparison with hard clustering methods, in which a pattern belongs to a single cluster, soft clustering algorithms allow patterns to belong to all clusters with varying degrees of membership. One of the most widely used soft clustering algorithms is the fuzzy  $c$ -means algorithm (henceforth FCM) (Bezdek et al., 1984). The FCM algorithm attempts to partition a finite set of  $n$  objects  $K = \{k_1, \dots, k_n\}$  into a collection of  $c$  fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of  $c$  cluster centers  $C = \{c_1, \dots, c_c\}$  and a partition matrix  $W = w_{i,j} \in [0, 1]$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, c$ , where each element  $w_{i,j}$  tells the degree to which element  $k_i$  belongs to cluster  $c_j$ . Like the k-means algorithm, the FCM aims to minimize an objective function, given as:

$$J_m(U, \beta) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m D_{ik}(x_k, \beta_i) \quad (4)$$

where

$u_{ik}$  is the membership of the  $k$ th object in the  $i$ th cluster;

$\beta_i$  represents the  $i$ th cluster prototype;

$m \geq 1$  is the degree of fuzziness;

$c \geq 2$  is the number of cluster;

$n$  represents the number of data points;

$D_{ik}(x_k, \beta_i)$  is the Euclidean distance between  $k^{th}$  object and  $i^{th}$  cluster center.

## 4 Evaluation

In this section, we discuss the extraction of evaluation sets for the validation of the clustering results. When a clustering solution has been obtained for a data set, it must also be presented in a manner which provides an overview of the content of each cluster. For that matter, we need an evaluation set that can provide class labels for each nominal *a priori*. The clustering task is then to

assign these nominals to a given number of clusters such that each cluster contains all and only those nominals that are members of the same class. Given the ground truth class labels, it is trivial to determine how accurate the clustering results are. This evaluation set is built using the Hindi wordnet<sup>4</sup> (Narayan et al., 2002), a lexical resource composed of synsets and semantic relations. Animacy of a nominal is taken from concept ontologies listed in the wordnet. We created two data sets using Hindi Wordnet:

- SET-1: This set contains nominals that are either animate or inanimate across senses listed in the wordnet. For example, nominals like *baalak* ‘boy’ with all senses animate and *patthar* ‘stone’ with all senses inanimate would fall under this set, whereas *kuttaa* ‘dog’ or ‘pawl’ with varying animacy across senses would not qualify to be included in this set. The sense hierarchies corresponding to animate (dog) and inanimate (pawl) senses of noun *kuttaa* are represented in Figure 1. There are 6039 nominals in this set. It is used to evaluate the results and determine the accuracy of clustering.
- SET-2: In this set all the nominals listed in wordnet are extracted irrespective of their animacy. There are around 7030 (SET-1+991) nominals in this set. It is used to evaluate the borderline cases with equal likelihood to fall in any cluster.

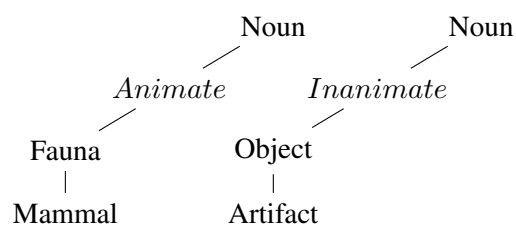


Figure 1: Animate and Inanimate Senses of noun *kuttaa*

It must be noted that only those nominals that satisfy the marked threshold of  $>10$  are considered, as discussed in Subsection 3.1.

## 5 Experiments and Results

In this section, we will discuss our clustering experiments followed by a thorough error analysis of

<sup>4</sup><http://www.cfilt.iitb.ac.in/wordnet/webhwn/wn.php>



	Animate			In-animate		
	Precision	Recall	F-score	Precision	Recall	F-score
Baseline	66.99	44.45	53.44	97.82	39.78	56.56
SVM	78.90	77.70	78.24	95.15	95.43	95.29
<b>Cmeans</b>	<b>57.8</b>	<b>89.18</b>	<b>70.0</b>	<b>97.3</b>	<b>85.65</b>	<b>91.15</b>
<b>Swedish</b>	81.9	64.0	71.8	96.4	98.6	97.5

Table 1: Comparison of Results.

the results achieved. In order to put our approach into perspective, we will first setup a baseline and establish a supervised benchmark for the task. For both the classification and clustering experiments (discussed shortly), we use SET-1. All the experiments are performed with the feature vectors representing the behavior of the corresponding nominals towards the case system of Hindi. The results are listed in Table 1.

### 5.1 Baseline

As discussed in Section 3, animates should occur more frequently with the case markers to the left of the Scale (2), while inanimates should occur more frequently with the ones to the right of the Scale. Thus, we used the frequency of a nominal with the case markers on the edges of the Scale (2), i.e., *ne* and *mem*, to set up the baseline. If a nominal occurs more frequently with *ne*, it is considered as animate, whereas if it occurs more frequently with *mem*, it is considered as inanimate. As the Table 1 shows, we could only achieve an average recall of 42 by this approach. This implies that the interaction of a nominal with the overall case system of a language, rather than an individual case marker, provides a better picture about its animacy.

### 5.2 Supervised Classification

For supervised classification, we used Support Vector Machines (SVMs). To train and test the SVM classifier, we used the LIBSVM package (Chang and Lin, 2011). We performed a 5-fold cross validation with a random 80-20 split of SET-1 for training and testing the classifier. The average accuracies are reported in Table 1. Although, the overall accuracy of supervised classification is higher, it comes with a cost of manual annotation of training data.

### 5.3 Clustering

A clustering experiment is performed with FCM clustering algorithm on SET-1 and SET-2, with

parameters  $c$  ‘number of clusters’ and  $m$  ‘degree of fuzziness’ set to 2. We used the  $F_\beta^5$  and *purity* to evaluate the accuracy of our clustering results, which are two widely used external clustering evaluation metrics (Manning et al., 2008). In order to evaluate the results, each nominal in SET-1 is assigned to the cluster  $j$  for which its cluster membership  $w_k^c$  (the degree of membership of a nominal  $k$  to cluster  $j$ ) is highest; i.e.,  $\text{argmax}_{c \in C} \{w_k^c\}$ . As shown in Table 2, the clustering solution by FCM has achieved  $F_\beta$  and purity scores of 89 and 86. Further, cluster 1 roughly corresponds to the Hindi wordnet inanimate class of nominals (86 recall) and cluster 2 corresponds to the Hindi wordnet animate class (89 recall). In (Øvrelid, 2009) and (Bowman and Chopra, 2012) animate nouns are reported as a difficult class to learn. The problem is attributed to the skewness in the training data. Animate nouns occur less frequently than inanimate nouns. In our clustering experiments, however, animates have shown higher predictability than inanimates. We have achieved a high recall on both animate as well as inanimate nominals. Further, we infer animacy of all types of nominals while (Øvrelid, 2009) and (Bowman and Chopra, 2012) have restricted the learning only for common noun lemmas. Furthermore, our method also identifies ambiguous nominals, as shown in Table 4. Although less feasible, we also present the results produced by Øvrelid (2009) ( $>10$ ) in Table 1 for a rough comparison.

Cluster	Animate	In-animate	$F_\beta$	Purity
1	117	4246	91	97
2	965	711	70	58
Total	1082	4957	89	86

Table 2: Clustering Results on SET-1.

As presented in Table 3, there are 828 instances

<sup>5</sup> $\beta$  is a coefficient of the relative strengths of precision and recall. We have set its value to 1, for all the results we have reported in this paper.

of wrong clustering. However, upon close inspection the clustering of these instances seems theoretically grounded, thus adding more weight to our results. We discuss these instances below:

1. **Personal Names:** As discussed in Section 3.1, personal names are ambiguous and can be used as common nouns with generic reference. Hindi Wordnet doesn't enlist personal names (except for very popular names), though their common usages are listed. For example the noun *baadal* 'cloud' is present in wordnet while its use as personal name is not listed. In the corpora used for the extraction of distributions, around 325 such nouns are actually used as personal names. Although, these nouns are correctly clustered as animates, they are evaluated as instances of wrong clustering, because of the inanimate sense they have in the Hindi Wordnet. This addresses the problem of **low precision** and **low purity** for animate nominals in our experiments. Similarly, the names used for gods, goddesses and spirits are also treated as inanimates in Hindi Wordnet. However, corpus distributions project them as animates due to their high ability to instigate an action. An example case that was wrongly clustered is *rab* 'God'.
2. **Lower Animates:** Although wordnet lists these nominals as animates which in fact they are, they are linguistically seen as inanimates and thus are clustered as such. In our experiments, *titli* 'butterfly' is clustered with inanimates.
3. **Natural Forces:** These nominals have a high control over an action and their distributions are more like higher animates. *bhuchaal* 'earthquake' is an instance of this over generalization.
4. **Psychological Nouns:** Nouns like *pare-shaanii* 'stress' are conceptualized as a force affecting us psychologically. These nominals are thus distributed like nominals of high control, which leads to an over generalization of these nouns as animates.
5. **Metonymies:** Nouns like country names, as discussed in Section 3.1, apart from referring to geographical places can also refer to

their inhabitants, teams, governments. Wordnet only treats these terms as inanimates (place). *Australia*, though treated as inanimate in Hindi Wordnet, is clustered with animates in our experiments.

6. **Machines:** A few cases of machines are also seen to be over generalized as animates. Machines show an animate like control (directly or indirectly) over an action.
7. **Nouns of Disability:** As these expressions refer to animates with some disability, they lack any control over an action and are distributed like inanimates. An example of this over generalization is noun *ghaayal* 'wounded'.
8. **Others:** These are actual instances of wrong clustering and as we noticed, these instances could probably be addressed by choosing an optimal frequency threshold to capture the unmarked (prototypical) behavior of a nominal. We have not addressed the tuning of this parameter in this work. However, we plan to take it up in future.

Nominal Type	Nominal Count
<i>Personal Name</i>	325
<i>Lower Animate</i>	104
<i>Natural Force</i>	67
<i>Psychological Nouns</i>	74
<i>Metonymies</i>	86
<i>Machine</i>	30
<i>Nouns of Disability</i>	44
<i>Others</i>	98
<b>Total</b>	<b>828</b>

Table 3: Error Classification on SET-1

In order to evaluate the ambiguous nominals that can have both animate and inanimate references in different contexts, we use SET-2. The borderline cases i.e, the nominals whose cluster membership score  $w_k^c$  is  $\sim 0.5$  are evaluated against the ambiguous nominals listed in SET-2. As shown in Table 4, from 991 ambiguous nominals 535 are clustered with inanimates in Cluster 1, while 439 cases are clustered with animates in Cluster 2. The fact that these nominals possess both the animate and inanimate senses, clustering them in either of the class should not be considered wrong. Although they have differing animacy as listed in Hindi Wordnet, probably they have

been used only in animate or inanimate sense in the corpora used in our experiments. Table 4 also shows that 187 nominals have a uniform distribution over the factors that discriminate animacy. Among these 150 nouns are listed as inanimate in Hindi Wordnet. Upon close inspection, these cases were found to be metonymies. As discussed earlier, Hindi Wordnet treats metonymies as inanimate, but in fact they are ambiguous. Thus our clustering of these nominals is justified.

Cluster	Animate	In-animate	Ambiguous
1	107	4149	535
2	955	658	439
$w_k^c \approx 0.5$	20	150	17
Total	1082	4957	991

Table 4: Clustering Results on SET-2

In Section 3, we stated that the distributions of nominals will be skewed on the control hierarchy. The results have clearly indicated that such skewness does in fact exist in the data, as shown in Figure 2. The cluster prototypes, returned by the fuzzy clustering, show animates are left skewed while inanimates are right skewed on the hierarchy of control. However, in our clustering experiments the order of dative/accusative and instrumental case markers on the control hierarchy (Scale 1) has been swapped. The dative/accusative case is more biased towards animates while instrumental case shows the reverse tendency. The reason for this is the ambiguity in these case markers. The instrumental case *se* mark roles such as cause, instrument, source and material. Among which cause and instrument imply high control while source and material imply a low control over an action. Almost 82% of instances of instrumental case depict a non-causal role while only 18% show a causal relation as annotated in the Hindi dependency treebank (Bhatt et al., 2009). Similarly, the dative/accusative case *ko* is used for experiencer subject, direct and indirect objects (Mohan, 1990, p. 72). Among these, only direct objects realized by definite inanimates are *ko* marked (Differential Object Marking), thus making it a more probable case marker for animates.

Before concluding the paper, we will discuss some of the issues related to the portability of our approach to other languages with rich morphological case. We will briefly discuss these issues below:

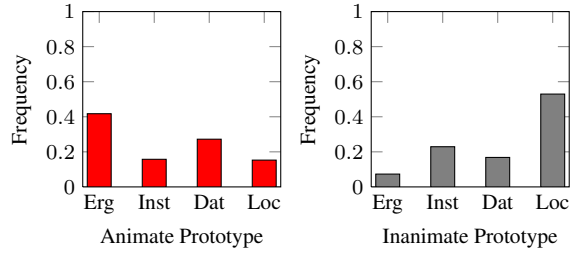


Figure 2: Skewed Marked Distribution of Cluster Prototypes.

- Case Ambiguity or Case Syncretism:** For an ideal performance, we expect a separate case marker for each individual case listed on Scale 1. Unfortunately, case markers are usually ambiguous. A case marker can have more than one case function in a language. In our work on Hindi, we saw that case ambiguity does have an impact on the results. We could afford to exclude highly ambiguous genitive case marker from our experiments (Mohan, 1990). However, how case ambiguity will impact the animacy prediction in other languages remains to be seen.
- Nominal Ambiguity:** As a matter of fact, animacy is an inherent and a non-varying property of nominal referents. However, due to lexical ambiguity (particularly metonymy), animacy of a word form may vary across contexts. We have addressed this problem by capturing the mixed membership of such ambiguous nominals. However, since animacy of a nominal is judged on the basis of its distribution, the animacy of an ambiguous nominal will be biased towards the sense with which it occurs in a corpora.
- Type of Morphology:** Case marking may be realized in different ways depending on the morphological type of a language. In case of inflectional and agglutinative languages, case markers, if present, are bound to a noun stem, while in analytical languages they are free morphemes usually lying adjacent to a nominal they mark. Although, the way case markers are realized may not affect the animacy prediction directly, it may impact the extraction of case marked distribution of nominals. Particularly, in case of agglutinative and inflectional languages extracting the multiple case marked word forms of a particular noun stem could be a challenging task.

## 6 Conclusion

In this work we report a technique to exploit the systematic correspondences between different linguistic phenomena to infer the important semantic category of animacy. The case marked distributions of nominals are clustered with fuzzy *cmeans* clustering into two clusters that approximate the binary dimensions of animacy. We achieved satisfactory results on the binary distinction of nominals on animacy. A  $F_\beta$  score of 89 and purity of 86 confirm efficiency of our approach. However, the performance of our system can be further improved by incorporating features from a dependency parser and an anaphora resolution system, as discussed in (Øvrelid, 2009).

In view of the Indo-Wordnet project (Bhattacharyya, 2010) that aims to build wordnet for major Indian languages, our approach can be used to predict animacy of nouns to leverage the cost and time associated with manual creation of such resources. Given the availability of large data on web for many Indian languages, our method can predict this information with satisfactory results. In the future, we also plan to explore the interaction between control and verb semantics, so as to classify verbs based on the amount of control required. This information can also be incorporated into the process of building Indo-wordnets.

## Acknowledgments

We would like to thank the anonymous reviewers for their useful comments which helped to improve this paper. We furthermore thank Sambhav Jain for his help and useful feedback.

## References

- Judith Aissen. 2003. Differential object marking: Iconicity vs. economy. *Natural Language & Linguistic Theory*, pages 435–483.
- Kirk Baker and Chris Brew. 2010. Multilingual animacy classification by sparse logistic regression. *Information Concerning OSDL OHIO STATE DISERTATIONS IN LINGUISTICS*, page 52.
- James C Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, pages 191–203.
- Akshar Bharati, Samar Husain, Bharat Ambati, Sambhav Jain, Dipti Sharma, and Rajeev Sangal. 2008. Two semantic features make all the difference in parsing accuracy. *Proceedings of International Conference on Natural Language Processing (ICON08)*.
- Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189. Association for Computational Linguistics.
- Pushpak Bhattacharyya. 2010. IndoWordNet.
- Samuel R Bowman and Harshit Chopra. 2012. Automatic animacy classification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 7–10. Association for Computational Linguistics.
- Holly P Branigan, Martin J Pickering, and Mikihiro Tanaka. 2008. Contributions of animacy to grammatical function assignment and word order during production. *Lingua*, 118(2):172–189.
- Joan Bresnan, Anna Cueni, Tatiana Nikitina, R Harald Baayen, et al. 2007. Predicting the dative alternation. *Cognitive foundations of interpretation*, pages 69–94.
- Miriam Butt and Tafseer Ahmed. 2011. The redevelopment of Indo-Aryan case systems from a lexical semantic perspective. *Morphology*, pages 545–572.
- Miriam Butt. 2006. The dative-ergative connection. *Empirical issues in syntax and semantics*, pages 69–92.
- Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, page 27.
- Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.
- William Croft. 2002. *Typology and universals*. Cambridge University Press.
- Peter De Swart, Monique Lamers, and Sander Lestrade. 2008. Animacy, argument structure, and argument encoding. *Lingua*, 118(2):131–140.
- Felice Dell’Orletta, Alessandro Lenci, Simonetta Montemagni, and Vito Pirrelli. 2005. Climbing the path to grammar: A maximum entropy model of subject/object learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 72–81. Association for Computational Linguistics.
- Richard Evans and Constantin Orasan. 2000. Improving anaphora resolution by identifying animate entities in texts. In *Proceedings of the Discourse Anaphora and Reference Resolution Conference (DAARC2000)*, pages 154–162.
- Christiane Fellbaum. 2010. *WordNet*. Springer.

- Paul J Hopper and Sandra A Thompson. 1980. Transitivity in grammar and discourse. *Language*, pages 251–299.
- Seppo Kittilä, Katja Västi, and Jussi Ylikoski. 2011. *Case, Animacy and Semantic Roles*. John Benjamins Publishing.
- Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press Cambridge.
- Colin P Masica. 1993. *The Indo-Aryan Languages*. Cambridge University Press.
- Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, pages 373–408.
- Tara Mohanan. 1990. *Arguments in Hindi*. Ph.D. thesis, Stanford University.
- Dipak Narayan, Debasri Chakrabarty, Prabhakar Pande, and Pushpak Bhattacharyya. 2002. An experience in building the indo wordnet-a wordnet for hindi. In *First International Conference on Global WordNet, Mysore, India*.
- Constantin Orasan and Richard Evans. 2007. Np animacy identification for anaphora resolution. *J. Artif. Intell. Res.(JAIR)*, 29:79–103.
- Constantin Orsan and Richard Evans. 2001. Learning to identify animate references. In *Proceedings of the 2001 workshop on Computational Natural Language Learning-Volume 7*, page 16. Association for Computational Linguistics.
- Lilja Øvrelid and Joakim Nivre. 2007. When word order and part-of-speech tags are not enough – Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 447–451.
- Lilja Øvrelid. 2006. Towards robust animacy classification using morphosyntactic distributional features. In *Proceedings of the 2006 Conference of the European Chapter of the Association for Computational Linguistics (EACL): Student Research Workshop*, pages 47–54.
- Lilja Øvrelid. 2009. Empirical evaluations of animacy annotation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Michael Silverstein. 1986. Hierarchy of features and ergativity. *Features and projections*, pages 163–232.

# The Complexity of Math Problems – Linguistic, or Computational?

Takuya Matsuzaki<sup>1</sup>, Hidenao Iwane<sup>2</sup>, Hirokazu Anai<sup>2,3</sup> and Noriko Arai<sup>1</sup>

<sup>1</sup> National Institute of Informatics, Japan

<sup>2</sup> Fujitsu Laboratories Ltd., Japan <sup>3</sup> Kyushu University, Japan

{takuya-matsuzaki, arai}@nii.ac.jp; {iwane, anai}@jp.fujitsu.com

## Abstract

We present a simple, logic-based architecture for solving math problems written in natural language. A problem is firstly translated to a logical form. It is then rewritten into the input language of a solver algorithm and finally the solver finds an answer. Such a clean decomposition of the task however does not come for free. First, despite its formality, math text still exploits the flexibility of natural language to convey its complex logical content succinctly. We propose a mechanism to fill the gap between the simple form and the complex meaning while adhering to the principle of compositionality. Second, since the input to the solver is derived by strictly following the text, it may require far more computation than those derived by a human, and may go beyond the capability of the current solvers.

Empirical study on Japanese university entrance examination problems showed positive results indicating the viability of the approach, which opens up a way towards a true end-to-end problem solving system through the synthesis of the advances in linguistics, NLP, and computer math.

## 1 Introduction

Development of an NLP system usually starts by decomposing the task into several sub-tasks. Such a modular design is mandatory not only for the reusability of the component technologies and the extensibility of the system, but also for the sound and steady advancement of the research field. Each module, however, has to attack its sub-task in isolation from the entirety of the task, usually with a quite limited form and amount of knowledge. The separated sub-task is hence not

necessarily easy even for human. This problem has been investigated in various directions, including the solutions to the error-cascading in pipeline models (Finkel et al., 2006; Roth and Yih, 2007, e.g.), the injection of knowledge into the processing modules (Koo et al., 2008; Pitler, 2012, e.g.), and the invention of a novel way of modularization (Bangalore and Joshi, 2010, e.g.).

In this paper, we present a simple pipeline architecture for natural language math problem solving, and investigate the issues regarding the separation of the semantic composition mechanism and the mathematical inference. Although the separation between these two may appear to be of different nature than the above-mentioned issues regarding the system modularization, as we will see later, the technical challenges there are also in the tension between the generality of an implemented theory as a reusable component, and its coverage over domain-specific phenomena.

In the system, a problem is analyzed with a Combinatory Categorical Grammar (Steedman, 2001) coupled with a semantic representation based on the Discourse Representation Theory (Kamp and Reyle, 1993) to derive a logical form. The logical form is then rewritten to the input language of a solver algorithm, such as specialized math algorithms and theorem provers. The solver finally finds an answer through inference.

Natural language problem solving in math and related domain is a classic AI task, which has served as a good test-bed for the integration of various AI technologies (Bobrow, 1964; Charniak, 1968; Gelb, 1971, e.g.). Besides its attraction as a pure intellectual challenge, it has direct applications to the natural language interface for the formal systems such as databases, theorem provers, and formal proof checkers. The necessity of the interaction between language understanding and backend solvers has been pointed out in some of the classic works and also in closely related works

**terms**  $t ::= v \mid f(t_1, \dots, t_k) \mid \Lambda v.t \mid \Lambda v.D$   
**conditions**  $C ::= P(t_1, \dots, t_k) \mid \neg D \mid D_1 \rightarrow D_2$   
**DRSs**  $D ::= (\{v_1, \dots, v_k\}, \{C_1, \dots, C_m\})$

Figure 1: Syntax of DRS

such as Winograd’s SHRDLU (1971). A clear separation of the two layers is, however, an essential property for a wide-coverage problem solving system since we can extend it in a modular fashion, by the enhancement of the solver or the addition of different types of solvers.

The research question in the current paper is thus summarized as follows:

1. Can we derive the logical form of the problems compositionally, with no intervention of mathematical inference, and how?
2. Can we solve such a direct translation of the text to a logical form with the current state-of-the-art automatic reasoning technology?

After a brief overview of the system pipeline (§3), we present a technique for capturing the dynamic properties of the syntax-semantics mapping in the math problem text, which, at first sight, seem to call for mathematical inference during the derivation of a logical form (§4). We then describe remaining issues we found so far in the semantic analysis of math problem text (§5). Finally, the viability of the approach is empirically evaluated on real math problems taken from university entrance examinations. In the evaluation, we apply a solver to the logical forms derived through manually annotated CCG derivations and DRSs on the problem text (§6). In the current paper, we thus exclusively focus on the formal aspect of the semantic analysis, setting aside the problem of its automation and disambiguation. The final section concludes the paper and gives future prospects including the automatic processing of the math text.

## 2 Preliminaries

### 2.1 Discourse Representation Structure

We use a variant of Discourse Representation Structure (DRS) (Kamp and Reyle, 1993) for the semantic representation. DRS has been developed for the formal analysis of various discourse phenomena, such as anaphora and quantifier scopes beyond a single sentence.

Fig. 1 shows the syntax of DRS used in this paper.<sup>1</sup> In the definitions,  $f$  and  $P$  respectively denote a function and a predicate symbol and  $v$  denotes a variable. The definition is slightly extended from that by van Eijck and Kamp (2011) for incorporating higher-order terms. A term of the form  $\Lambda v.M$  denotes lambda abstraction in the object language, which is used to represent (mathematical) functions and sets<sup>2</sup>; we reserve  $\lambda$  for denoting the abstraction over DRSs (and terms) for the composition of DRSs. We define the interpretation of a DRS  $D$  indirectly through its translation  $D^\circ$  to a (higher-order) predicate logic as in Fig. 2.

As defined in Fig. 2, a DRS  $D = (\mathbf{V}, \mathbf{C})$  is basically interpreted as a conjunction of the conditions in  $\mathbf{C}$  that is quantified existentially by all the variables in  $\mathbf{V}$ . However, as in the second clause in Fig. 2, the variables in the antecedent of an implication are universally quantified and their scopes also cover the succedent; this definition is utilized in the analysis of sentences including indefinite NPs, such as donkey sentences.

The mechanism of the DRS composition in this paper is based on the formulation by van Eijck and Kamp (2011). They use an operation called merge (denoted by  $\bullet$ ) to combine two DRSs. Assuming no conflicts of variable names, it can be defined as:  $(\mathbf{V}_1, \mathbf{C}_1) \bullet (\mathbf{V}_2, \mathbf{C}_2) := (\mathbf{V}_1 \cup \mathbf{V}_2, \mathbf{C}_1 \cup \mathbf{C}_2)$ . Roughly speaking, this operation amounts to form the conjunction of the conditions in  $\mathbf{C}_1$  and  $\mathbf{C}_2$  allowing the conditions in  $\mathbf{C}_2$  to refer to the variables in  $\mathbf{V}_1$ . Consider the following discourse:

- $s_1$ : A monkey <sup>$x$</sup>  is sleeping.  
 $s_2$ : It <sub>$x$</sub>  holds a banana.

Assuming the anaphoric relation indicated by the super/sub-scripts, we have their DRSs as follows:

$$D_1 = (\{x\}, \{\text{monkey}(x), \text{sleep}(x)\})$$

$$D_2 = (\{y\}, \{\text{banana}(y), \text{hold}(x, y)\})$$

By merging them, we have

$$D_1 \bullet D_2 = \left( \{x, y\}, \left\{ \begin{array}{l} \text{monkey}(x), \text{sleep}(x), \\ \text{banana}(y), \text{hold}(x, y) \end{array} \right\} \right),$$

which is translated to  $\exists x. \exists y. (\text{monkey}(x) \wedge \dots \wedge \text{hold}(x, y))$  as expected.

<sup>1</sup>Disjunction can be defined by using implication and negation:  $D_1 \vee D_2 := (\{\}, \{\neg D_1\}) \rightarrow D_2$ .

<sup>2</sup>We represent the application of a  $\Lambda$ -term to another term, such as  $(\Lambda x.D)t$  and  $(\Lambda x.t_1)t_2$ , either by a special predicate  $\text{App}(f, x) \equiv fx$  or a function  $\text{app}(f, x) := fx$  according to the type of  $f$ . Compound terms of the form  $t_1 t_2$  are hence not in the definitions.

$$\begin{array}{lll}
\text{Assuming } D_1 = (\{v_1, \dots, v_k\}, \{C_1, \dots, C_m\}), & (\neg D)^\circ := \neg D^\circ & (\Lambda v.D)^\circ := \Lambda v.(D^\circ) \\
D_1^\circ := \exists v_1 \dots \exists v_k. (C_1^\circ \wedge \dots \wedge C_k^\circ) & (P(t_1, t_2, \dots))^\circ := P(t_1^\circ, t_2^\circ, \dots) & (\Lambda v.t)^\circ := \Lambda v.(t^\circ) \\
(D_1 \rightarrow D_2)^\circ := \forall v_1 \dots \forall v_k. ((C_1^\circ \wedge \dots \wedge C_m^\circ) \rightarrow D_2^\circ) & (f(t_1, t_2, \dots))^\circ := f(t_1^\circ, t_2^\circ, \dots) & v^\circ := v
\end{array}$$

Figure 2: Translation of DRS to HOPL

$$\begin{array}{c}
\frac{\text{When}}{S/S/S} > \frac{\frac{\text{the centers of } C_1 \text{ and } C_2}{S/(S \setminus NP)} \quad \frac{\text{coincide}}{S \setminus NP}}{\lambda P.(\{x, x_1, x_2\}, \{x = [x_1, x_2], x_1 = \text{center.of}(C_1), x_2 = \text{center.of}(C_2)\}) \bullet P x} : \lambda x.(\{\}, \{\text{coincide}(x)\})} \\
: \lambda P. \lambda Q. P \rightarrow Q > \frac{S : (\{x, x_1, x_2\}, \{x = [x_1, x_2], x_1 = \text{center.of}(C_1), x_2 = \text{center.of}(C_2), \text{coincide}(x)\})}{S/S : \lambda Q.(\{x, x_1, x_2\}, \{x = [x_1, x_2], x_1 = \text{center.of}(C_1), x_2 = \text{center.of}(C_2), \text{coincide}(x)\}) \rightarrow Q}
\end{array}$$

Figure 3: A part of CCG derivation tree

$$> \frac{X/Y : f \quad Y : a}{X : fa} \quad >B \frac{X/Y : f \quad Y/Z : g}{X/Z : \lambda x.f(gx)}$$

Figure 4: Example of combinatory rules

## 2.2 Combinatory Categorical Grammar

Combinatory Categorical Grammar (CCG) (Steedman, 2001) is a lexicalized grammar formalism. In CCG, the association between a word  $w$  and its syntactic/semantic property is specified by a lexical entry of the form  $w := C : S$ , where  $C$  is the category of  $w$  and  $S$  is the semantic interpretation of  $w$ . A category is either a basic category (e.g., S, N, NP) or a complex category of the form  $X/Y$  or  $X \setminus Y$ . For instance, we can assign the following categories and semantic interpretations to the region notation “[0, +∞)” and a bare noun phrase “positive number”:

$$\begin{array}{l}
[0, +\infty) := \text{NP} : \Lambda x.(\{\}, \{x \geq 0\}) \\
\text{positive number} := \text{N} : \lambda x.(\{\}, \{x > 0\})
\end{array}$$

since the region notation behaves as a proper noun and it can be represented by its characteristic function, while “positive number” functions like a common noun (recall that  $\Lambda$  is for the abstraction in the object language and  $\lambda$  stands for the abstraction for the DRS composition). A handful of combinatory rules define how the categories and the semantic interpretations of constituents are combined to derive a larger phrase. Fig. 4 shows two of the rules. A part of a derivation tree for “When the centers of  $C_1$  and  $C_2$  coincide” is shown in Fig. 3. As shown in the figure, the semantic representation in DRS is composed by the beta reduction and the DRS merge operation. As we will see in §4, there are certain types of discourse for which the basic DRS composition machinery described so far does not suffice. We will return to this after a brief description of the whole system.

## 3 A Simple Pipeline for Natural Language Math Problem Solving

The main result in the current paper is a mechanism of semantic composition and an empirical support for our overall design choice. Although the NLP modules for the automatic processing and disambiguation are still under development, we show a brief overview of the whole system to give a clear image on the different representations of a problem at different stages of the pipeline.

**From text to logical form** The system receives a problem text with L<sup>A</sup>T<sub>E</sub>X-style markup on the symbolic mathematical expressions: e.g.,

Let  $a > 0$ ,  $b \leq 0$ , and  $0 < p < 1$ .  
 $\$P(p, p^2)\$$  is on the graph of the  
function  $y = ax - bx^2$ . Write  $b$  in  
terms of  $a$  and  $p$ .

We process the mathematical expressions with a symbolic expression analyzer and produce their possible interpretations as lexical entries. For instance,  $y = ax - bx^2$  in the above example will receive at least two interpretations:

$$\begin{array}{l}
\$y = ax - bx^2\$ := S : (\{\}, \{y = ax - bx^2\}) \\
\$y = ax - bx^2\$ := \text{NP} : \Lambda x.ax - bx^2.
\end{array}$$

The first lexical entry is for the usages such as “Hence  $y = ax - bx^2$ ,” where the expression denotes a proposition and behaves as a sentence. The second entry is for the usage as a noun phrase as in the example, which stands for a function.

We add such dynamically generated lexical entries to the lexicon and then analyze the sentences with a CCG parser. From the resulting CCG derivation trees, we will obtain a DRS for each sentence. For the above example, we will have the following DRSs (the third one is in the extended



language we'll introduce in the next section):

$$\begin{aligned} D_1 &= (\{\}, \{a > 0, b \leq 0, 0 < p, p < 1\}) \\ D_2 &= (\{\}, \{P = (p, p^2), \text{on}(P, \Lambda x.ax - bx^2)\}) \\ D_3 &= \text{Find}(b') [cc; \exists^{-1}a; \exists^{-1}p; b = b'] \end{aligned}$$

A discourse structure analyzer receives the DRSs and determines the logical relations among them while selecting an antecedent for each anaphoric expression. The net result of this stage is a large DRS that represents the whole problem. For the above example, we have their sequencing as the result:  $D_1; D_2; D_3$ . The sequencing operator (;) basically means conjunction (merge) of the DRSs, but it is also used to connect the meanings of a declarative sentence and an imperative sentence. The large DRS is then translated by a process defined in the next section, giving a HOPL formula enclosed by a directive to the solver:

$$\text{Find}(b') \left[ \begin{array}{l} a > 0 \wedge b \leq 0 \wedge 0 < p \wedge p < 1 \wedge \\ \exists P. \left( \begin{array}{l} P = (p, p^2) \wedge \\ \text{on}(P, \Lambda x.ax - bx^2) \wedge b = b' \end{array} \right) \end{array} \right],$$

where  $\text{Find}(v)[\phi]$  is a directive to find the value of variable  $v$  that satisfies the condition  $\phi$ .

**From logical form to solver input** Many of the current automatic reasoners operate on first-order formulas. To utilize them, we hence have to transform the HOPL formula in a directive to an equivalent first-order formula. Such transformation is of course not possible in general. However, we found that a greedy rewriting procedure suffices for that purpose on all of the high-school level math problems used in the experiment.

In the rewriting procedure, we iteratively apply several equivalence-preserving transformations including the beta-reduction of  $\Lambda$ -terms and rewriting of the predicates and functions using their definitions. For the above example, by using some trivial simplifications and the definition of  $\text{on}(\cdot, \cdot)$ :

$$\forall x. \forall y. \forall f. (\text{on}((x, y), f) \leftrightarrow (y = fx)),$$

we have the following directive holding a first-order formula:

$$\text{Find}(b') \left[ \begin{array}{l} a > 0 \wedge b \leq 0 \wedge 0 < p \wedge p < 1 \wedge \\ p^2 = ap - bp^2 \wedge b = b' \end{array} \right].$$

**Solver Algorithms** In addition to the generic first-order theorem provers, we can use specific algorithms as the solver when the formula is expressible in certain theories. Among them, many

mathematical and engineering problems can be naturally translated to formulas consisting of polynomial equations, inequalities, quantifiers ( $\forall, \exists$ ) and boolean operators ( $\wedge, \vee, \neg, \rightarrow$ , etc). Such formulas construct sentences in the first-order theory of real closed fields (RCF).

In his celebrated work, Tarski (1951) showed that RCF allows quantifier-elimination (QE): for any RCF formula  $\phi(x_1, \dots, x_n)$ , there exists an equivalent quantifier-free formula  $\psi(x_1, \dots, x_n)$  in the same vocabulary. For example, the formula  $\exists x.(x^2 + ax + b \leq c)$  can be reduced to a quantifier-free formula  $a^2 - 4b + 4c \geq 0$  by QE.

Automated theorem proving is usually very costly. For example, QE for RCF is doubly exponential on the number of quantifier alternations in the input formula. The problems containing only six variables may be hard for today's computer with the best algorithm known. However, several positive results have been attained as the result of extensive search for practical algorithms during the last decades (see (Caviness and Johnson, 1998)). Efficient software systems of QE have been developed on several computer algebra systems, such as SyNRAC (Iwane et al., 2013).

## 4 Formal Analysis of Math Problem Text

In this section, we first summarize the most prominent issues we found so far in the linguistic analysis of high-school/college level math problems and then present a solution.

### 4.1 Problems

**Context-dependent meanings of superlatives and their alike** The meaning of a superlatives and semantically similar expressions such as "maximum" generally depends highly on the context. For example, the interpretation of "John was the tallest" depends on the group (of people) that is prominent in the discourse:

There were ten boys. John was the tallest.

This context-dependency can be made more explicit by paraphrasing it to a comparative (Heim, 2000): "John was taller than *anyone else*," where "anyone else" refers, depending on the context, to the group against which John was compared.

In math text, however, we can usually determine the range of the "anyone else" without ambiguity:

Assume  $a + b = 3$ . Find the maximum value of  $ab$ .

Here, the set of values that should be compared against the maximum value is, with no ambiguity, all the possible values of  $ab$  that is determined by the preceding context. Once we have a representation of such a set, it is easy to write the semantic interpretation of the phrase “maximum value of  $\alpha$ .” But, how can we obtain a representation of such a set without inference?

**Discrimination between free/bound variable**  
We can explicitly specify that a variable should be interpreted as being free, as in:

Let  $R$  be a square with perimeter  $l$ .  
Write the area of  $R$  in terms of  $l$ .

This discourse may be translated to

$$\text{Find}(a) \left[ \exists R. \left( \begin{array}{l} \text{is\_square}(R) \wedge \\ \text{perimeter\_of}(R) = l \wedge \\ \text{area\_of}(R) = a \end{array} \right) \right]$$

but not to

$$\text{Find}(a) \left[ \exists R. \exists l. \left( \begin{array}{l} \text{is\_square}(R) \wedge \\ \text{perimeter\_of}(R) = l \wedge \\ \text{area\_of}(R) = a \end{array} \right) \right]$$

since, assuming the proper definitions of the functions and predicates, the first one is equivalent to  $\text{Find}(a)[a = l^2/16]$  but the second one is equivalent to  $\text{Find}(a)[a > 0]$ . How can we specify a variable be *not* bound?

**Imperatives** Math problems usually include imperatives such as “Find/Write...,” and “Prove/Show...”. How can we derive correct interpretations of those imperatives, which depend on the semantic content of preceding declarative sentences, but are not a part of the declarative meaning of a discourse?

## 4.2 Solution by iDRS

Although the above-mentioned phenomena are quite common in math problem text, we found it is difficult to derive the meanings of such expressions within the basic compositional DRS framework introduced in §2. All of the examples above involve the manipulation and modification of the context in a discourse.

We present an extension of the DRS composition mechanism that covers expressions like the above examples. The basic idea is to introduce another layer of semantic representation called iDRS hereafter, which provides a device to manipulate

**terms**  $t ::= v \mid f(t_1, \dots, t_k) \mid \Lambda v.t \mid \Lambda v.I$

**iDRS**  $I ::= P(t_1, \dots, t_k) \mid \neg I \mid I_1 \rightarrow I_2 \mid \exists v \mid I_1; I_2 \mid \exists^{-1}v \mid \text{Find}(v)[I] \mid \text{Show}[I] \mid cc$

Figure 5: Syntax of iDRS

the representation of the preceding context during the semantic composition.<sup>3</sup>

First we define the syntax of iDRS as in Fig. 5. In the definition, the variables  $P, f, t$ , and  $v$  follows the same convention as in the DRS definition. In words, an iDRS represents either a DRS condition (the first row of the definition of  $I$ ), a quantification  $\exists v$ , which corresponds to a DRS having only one variable,  $(\{v\}, \{\})$ , a sequencing  $I_1; I_2$  of two iDRSs, or the new ingredients in the rest of the definition that will be explained shortly.

The “anti-quantifier”  $\exists^{-1}v$  means an operation that cancels the quantification on  $v$  that precedes  $\exists^{-1}v$ .  $\text{Find}(v)[I]$  is a directive that requires to find the set of the values of variable  $v$  which satisfy the condition represented by  $I$ . Similarly,  $\text{Show}[I]$  is a directive that requires to prove the statement represented by  $I$ . Note that these two directives are not specific to any solvers; The choice of the solver depends on the theory (e.g., RCF) under which the formula in a directive is understood. The last element,  $cc$ , can be considered as a special ‘variable’, through which we can always retrieve an iDRS representation of the context that precedes the position marked by the  $cc$ .

Using these new ingredients, we can now write, for instance, the semantic representation of the phrase “maximum value” as follows:

$$\text{N/NP}_{\text{of}} : \lambda x. \lambda m. \max(\Lambda y. (cc; y = x), m),$$

assuming that the two-place predicate  $\max(s, m)$  is defined to be true iff  $m$  is the maximum element in the set  $s$  (represented by a  $\Lambda$ -term). A sentence “the maximum value of  $x$  is  $m$ ” will thus have  $\max(\Lambda y. (cc; y = x), m)$  as its semantic representation, which means that  $m$  is the maximum value of  $x$  that satisfies the condition specified by the preceding context.

<sup>3</sup>This approach shares much with a kind of dynamic semantics such as those by Bekki (2000) and Brasoveanu (2012), in which a representation of the context can also be accessed in the semantic language. An important difference is that in their approaches the context is represented as a set of assignment functions, while we represent them directly as an iDRS. This difference is crucial for our purpose since we eventually need to obtain a (first-order) formula on which an automatic reasoner operates.

$\{\{I_1; I_2\}\}_c := \{\{I_1\}\}_c; \{\{I_2\}\}_{c; [I_1]_c}$	$\llbracket cc; I \rrbracket_c := c; \llbracket I \rrbracket_c$	$\llbracket P(t_1, \dots) \rrbracket_c := P(\llbracket t_1 \rrbracket_c, \dots)$
$\{\{\text{Find}(v)[I]\}\}_c := \text{Find}(v) \llbracket [I]_c \rrbracket$	$\llbracket cc \rrbracket_c := c$	$\llbracket \exists v \rrbracket_c := \exists v$
$\{\{\text{Show}[I]\}\}_c := \text{Show} \llbracket [I]_c \rrbracket$	$\llbracket I_1; I_2 \rrbracket_c := \llbracket I_1 \rrbracket_c; \llbracket I_2 \rrbracket_{c; [I_1]_c}$	$\llbracket \exists^{-1} v \rrbracket_c := \exists^{-1} v$
$\{\{I_1 \rightarrow I_2\}\}_c := \{\{I_2\}\}_{c; [I_1]_c}$	$\llbracket I_1 \rightarrow I_2 \rrbracket_c := \llbracket I_1 \rrbracket_c \rightarrow \llbracket I_2 \rrbracket_{c; [I_1]_c}$	$\llbracket v \rrbracket_c := v$
$\{\{I\}\}_c := \epsilon$	$\llbracket \neg I \rrbracket_c := \neg \llbracket I \rrbracket_c$	$\llbracket f(t_1, \dots) \rrbracket_c := f(\llbracket t_1 \rrbracket_c, \dots)$
	$\llbracket \text{Find}(v)[I] \rrbracket_c := \exists v; \llbracket I \rrbracket_c$	$\llbracket \Lambda v. t \rrbracket_c := \Lambda v. \llbracket t \rrbracket_c$
	$\llbracket \text{Show}[I] \rrbracket_c := \llbracket I \rrbracket_c$	$\llbracket \Lambda v. I \rrbracket_c := \Lambda v. \llbracket I \rrbracket_c$

Figure 6: Transformation from iDRS to directive sequence

Let's take the following problem as an example:

Let  $p > 0$ .  $R$  is a rectangle whose perimeter is  $p$ . Find the maximum value of the area of  $R$  as a function of  $p$ .

We have its iDRS representation shown below, by parsing the sentences and composing the resulting iDRSs into one (in this case, just by sequencing the three sentences' iDRSs):

$$\left[ \begin{array}{l} 0 < p; \\ \text{is\_rectangle}(R); \text{perimeter\_of}(R) = p; \\ \exists m; \max(\Lambda x. [cc; x = \text{area\_of}(R)], m); \\ \text{Find}(a)[cc; \exists^{-1} p; a = m] \end{array} \right]$$

We then bind all free variables in the iDRS at their narrowest scopes:

$$\left[ \begin{array}{l} \exists p; 0 < p; \\ \exists R; \text{is\_rectangle}(R); \text{perimeter\_of}(R) = p; \\ \exists m; \max(\Lambda x. [cc; x = \text{area\_of}(R)], m); \\ \text{Find}(a)[cc; \exists^{-1} p; a = m] \end{array} \right]$$

This amounts to assume each variable appearing in a problem text is, unless it is explicitly quantified, interpreted to be existentially quantified as default, and to be universally quantified if it appears in the antecedent of an implication.

The iDRS is then processed by the functions  $\{\{\cdot\}\}_c$  and  $\llbracket \cdot \rrbracket_c$  defined in Fig. 6. In the definition,  $\epsilon$  stands for an empty sequence. The function  $\{\{\cdot\}\}_c$  extracts the imperative meaning from an iDRS, using  $\llbracket \cdot \rrbracket_c$  as a 'sub-routine' that extracts the declarative meaning from an iDRS. The suffix  $(c)$  of the two functions stands for the preceding context represented as an iDRS. When  $\llbracket \cdot \rrbracket_c$  processes a sequence  $I_1; I_2$  or an implication  $I_1 \rightarrow I_2$ , the declarative content of  $I_1$  (i.e.,  $\llbracket I_1 \rrbracket_c$ ) is appended to the preceding context  $c$ , and  $c; \llbracket I_1 \rrbracket_c$  is passed as the preceding context when processing  $I_2$ . When  $\llbracket \cdot \rrbracket_c$  finds a  $cc$  variable, it substitutes the  $cc$  with the current context stored in the suffix.

By applying  $\{\{\cdot\}\}_\epsilon$  to the iDRS of a problem, we can extract the logical form of the problem as a

sequence of directives. For the example problem, we have a single directive as follows:

$$\text{Find}(a) \left[ \begin{array}{l} \exists p; 0 < p; \\ \exists R; \text{is\_rectangle}(R); \text{perimeter\_of}(R) = p; \\ \exists m; \max(\Lambda x. \left[ \begin{array}{l} \exists p; 0 < p; \\ \exists R; \text{is\_rectangle}(R); \\ \text{perimeter\_of}(R) = p; \\ \exists m; x = \text{area\_of}(R) \end{array} \right], m); \\ \exists^{-1} p; a = m \end{array} \right]$$

Now, by the definition of  $\{\{\cdot\}\}_c$  and  $\llbracket \cdot \rrbracket_c$ , the iDRS  $I$  inside a directive  $\text{Find}(v)[I]$  or  $\text{Show}[I]$  includes only those elements that have a counterpart in the basic DRS except for the "anti-quantifiers." We can hence convert it to a HOPL formula, by first canceling the quantifications  $\exists v$  that precede  $\exists^{-1} v$  (i.e., deleting all occurrences of  $\exists v$  that appear before an occurrence of  $\exists^{-1} v$  in the iDRS, and deleting  $\exists^{-1} v$  itself), then converting it to a DRS by replacing the sequencing operator ';' to the merge operator, and finally translating it to a HOPL formula according to Fig. 2.

## 5 Remaining Issues in the Semantic Analysis of Math Problem Text

The mechanism presented in §4 significantly enhanced the coverage of the analysis over real problems. We however found several phenomena that can not be handled now.

**Free/bound variable distinction without a cue phrase** We have presented a mechanism to 'un-bind' the variables specified by a cue phrase, such as "(find  $x$ ) in terms of ( $y$ )." Some types of variables however have to be left free even without any explicit indication, e.g.:

Let  $p > 0$ . Find the area of a circle with radius  $p$ , centered at the origin.

Assuming  $\text{circle}(x, y, r)$  denotes a circle with radius  $r$  and centered at  $(x, y)$ , we want to derive

$$\text{Find}(a) [p > 0; a = \text{area\_of}(\text{circle}(0, 0, p))],$$

but our default variable binding rule gives

$\text{Find}(a) [\exists p; p > 0; a = \text{area\_of}(\text{circle}(0, 0, p))]$ .

This directive means to find the range of the areas of the circles with arbitrary radii, which is apparently not a possible reading of the problem. We found such cases in 3 out of the 32 test problems used in the experiment shown later.

**Scope inversion by a cue phrase** The hierarchy of the quantifier scopes in math text mostly follows the linear order of the appearance of the variables (either overtly quantified or not). This general rule can however be superseded by the effect of a cue phrase, as shown in the example problem and its possible translation in Fig. 7. In the figure, the formula inside the Show-directive mostly follows the discourse structure, in that the predicates from the first and the second sentence respectively form the antecedent and the succedent of the implication. The quantification on  $F$  is however dislocated from its default scope, i.e., the succedent, and moved to the outset of the formula by the effect of the underlined cue phrases. To handle such cases correctly, we would need a more involved mechanism for the manipulation of the context representation through the *cc* variable.

**Idiomatic expressions** As in other text genres, idiomatic multiword expressions are also problematic as can be seen in the following example:

By choosing  $x$  sufficiently large,  $y = 1/x$  can be made as close to 0 as desired.

As the example shows, a set phrase involving complex syntactic relations, e.g., “can do X as Y as desired by choosing Z sufficiently W” and “X approaches Y as Z approaches W,” can convey idiomatic meanings in math.

## 6 Empirical Results

We tested the feasibility of our approach on a set of problems selected from Japanese university entrance exams. Specifically, we wanted 1) to test the coverage of the semantic composition mechanism presented in §4 on real problems, and 2) to verify that there is no significant loss in the capability of the system due to the additional computational cost incurred by the separation of the semantic analysis from the mathematical reasoning.

The second point was confirmed by providing the ideal (100% correct) output from the

(forthcoming) NLP components to a state-of-the-art automatic reasoner and comparing the result against the performance of the reasoner on the input formulated by a human expert. Specifically, we manually gave the semantic representations of the problems as iDRSs or CCG derivation trees, and then automatically rewrote them into the language of RCF. The resulting formulas were fed to a solver to see whether the answers be returned in a realistic amount of time (30 seconds). The solver was implemented on SyNRAC (Iwane et al., 2013), which is an RCF-QE solver implemented as an add-on to Maple, and the (in)equation solving commands of Maple.

The problems were taken from the entrance exams of five first-tier universities in Japan (Tokyo U., Kyoto U., Osaka U., Kyushu U., and Hokkaido U.) for fiscal year 2001, 2003, 2005, 2007, 2009 and 2011. There were 249 problems in total. From them, we first eliminated those that included almost no natural language text, such like calculation problems. We then chose, from the remaining non-straightforward word problems, all the problems which could be solved with SyNRAC and Maple when the input was formulated by an expert of computer algebra. The formulation by an expert was done, of course, with no manual calculation, but otherwise it was freely done including the division of the solving process into several steps of QE and (in)equation solving.

As the result of that, we got 32 test problems, each of which contained 3.9 sentences on average. They include problems on algebra (of real and complex numbers), 3D and 2D geometry, calculus, and their combinations. For analyzing the result in more detail, we divided the problems into 78 sub-problems for which the correctness of the answers can be judged independently.

### 6.1 From discourse analysis to the solution

For the first experiment, we manually encoded the problems in the form of iDRSs. Each sentence in a problem was first encoded as a single iDRS, and the sentence-level iDRSs were combined (again manually) into a problem-level iDRS using the connectives defined in the iDRS syntax. In the manual encoding, the granularity of the representation, i.e., the smallest units of the semantic representation, was kept at the level of the actual words in the text whenever possible, intending that the resulting iDRSs closely match the representation

Problem: Point  $P$  is on the circle  $x^2 + y^2 = 4$  and  $l_P$  is the normal line to the circle at  $P$ . Show that  $l_P$  passes through a fixed point  $F$  irrespective of  $P$ .

$$\text{Show } \left[ \exists F. \left( \forall P. \forall l_P. \left( \left( \begin{array}{l} P \text{ is on } x^2 + y^2 = 4 \text{ and} \\ l_P \text{ is the normal line to the circle at } P \end{array} \right) \rightarrow l_P \text{ passes through } F \right) \right) \right]$$

Figure 7: Scope inversion by cue phrases

Let  $O(0, 0)$ ,  $A(2, 6)$ ,  $B(3, 4)$  be 3 points on the coordinate plane. Draw the perpendicular to line  $AB$  through  $O$ , which meets  $AB$  at  $C$ . Let  $s, t$  be real numbers, and let  $P$  be such that  $\overrightarrow{OP} = s\overrightarrow{OA} + t\overrightarrow{OB}$ . Answer the following questions.

(1) Calculate the coordinates of point  $C$ , and write  $|\overrightarrow{CP}|^2$  in terms of  $s$  and  $t$ .

(2) Let  $s$  be constant, and let  $t$  vary in the range  $t \geq 0$ . Calculate the minimum of  $|\overrightarrow{CP}|^2$ .

Figure 8: Kyushu University 2009 (Science Course) Problem 1

composed from word-level semantic representations. In the iDRS encoding of the 32 problems, the context-fetching mechanism through ‘*cc*’ variable was needed in 15 problems and the canceling of quantification was needed in 6 problems. These mechanisms thus significantly enhanced the coverage of the semantic composition machinery.

After rewriting the iDRSs to RCF formulas<sup>4</sup>, we fed them to the solver and got perfect answers for 19 out of the 32 problems. Out of the 78 sub-problems, 56 sub-problems (72%) were successfully solved. 12% of the sub-problems (9 sub-problems) failed due to the timeout in the QE solver. Besides the timeout, a major cause of the failures (7 sub-problems) was the fractional power (mainly square root) in the formula. Although we can mechanically erase the fractional powers to get an RCF formula, it was not implemented in the solver.<sup>5</sup> The remaining 6 sub-problems needed the free/bound variable distinction without any cue phrase (§5). Although half of them could be solved by manually specifying the free variables, we did not count them as solved here.

## 6.2 From syntactic analysis to the answer

We chose 14 problems from the 19 problems which were fully solved with the iDRS encod-

<sup>4</sup>The knowledge-base used to rewrite the HOPL formulas to first-order RCF formulas included 230 axioms for 86 predicates and 98 functions.

<sup>5</sup>In the formulation by the human expert, the use of square roots were avoided by encoding the conditions differently (e.g.,  $x \geq 0 \wedge x^2 = 2$  instead of  $\sqrt{x} = 2$ ).

ings. We manually analyzed the text following the CCG-based analyses of basic Japanese constructions given by Bekki (2010). We annotated the 44 sentences in the 14 problems with full CCG derivation trees and anaphoric links. We selected the 14 problems so that they cover different types of grammatical phenomena as much as possible. The final CCG lexicon contained 240 lexical entries (109 for function words and the rest for content words). The iDRS representations were then derived by (automatically) composing the semantic representations of the words according to the derivation trees and combining the sentence-level iDRSs to a problem-level iDRS as in the first experiment. Out of the 14 problems, we got fully correct answers for 13 problems. In the 14 problems, there were 33 sub-problems and we got correct answers for 32 of them; On only one sub-problem, the solver could not return an answer within the time limit. Fig. 8 shows an English translation of one of the 13 problems successfully solved with the CCG derivation trees as the input.

Overall, the results on the real exam problems were very promising: 72% of the sub-problems were successfully solved with the formula derived from a sentence-by-sentence, direct encoding of the problem. The experiment with manually annotated CCG derivation trees further showed that there was almost no additional cost introduced by the mechanical derivation of the logical forms from the word-level semantic representations.

## 7 Conclusion and Prospects

We have presented a logic-based architecture for automatic problem solving. The experiments on the university entrance exams showed positive results indicating the viability of the modular design.

Future work includes the development of the processing modules, i.e., the symbolic expression analyzer, the parser, and the discourse structure analyzer. Another future work is to incorporate different types of solvers to the system for covering a wider range of problems, with the ability to choose a solver based on the content of a problem.

## References

- Srinivas Bangalore and Aravind K. Joshi. 2010. *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*. Bradford Books. MIT Press.
- Daisuke Bekki. 2000. *Typed Dynamic Logic for Compositional Grammar*. Ph.D. thesis, University of Tokyo.
- Daisuke Bekki. 2010. *Formal Theory of Japanese Syntax*. Kuroshio Shuppan. (In Japanese).
- Daniel Gureasko Bobrow. 1964. *Natural language input for a computer problem solving system*. Ph.D. thesis, Massachusetts Institute of Technology.
- Adrian Brasoveanu. 2012. The grammar of quantification and the fine structure of interpretation contexts. *Synthese*, pages 1–51.
- Bob F. Caviness and Jeremy R. Johnson, editors. 1998. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer-Verlag, New York.
- Eugene Charniak. 1968. Carps: a program which solves calculus word problems. Technical report, Massachusetts Institute of Technology.
- Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. 2006. Solving the problem of cascading errors: approximate bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06*, pages 618–626, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jack P. Gelb. 1971. Experiments with a natural language problem-solving system. In *Proceedings of the 2nd international joint conference on Artificial intelligence, IJCAI'71*, pages 455–462, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Irene Heim. 2000. Degree operators and scope. In *Proceedings of Semantics and Linguistic Theory 10*, pages 40–64. CLC Publications.
- Hidenao Iwane, Hitoshi Yanami, Hirokazu Anai, and Kazuhiro Yokoyama. 2013. An effective implementation of symbolic-numeric cylindrical algebraic decomposition for quantifier elimination. *Theoretical Computer Science*. (in press).
- Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Studies in Linguistics and Philosophy. Kluwer Academic.
- Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio, June. Association for Computational Linguistics.
- Emily Pitler. 2012. Attacking parsing bottlenecks with unlabeled data and relevant factorizations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 768–776, Jeju Island, Korea, July. Association for Computational Linguistics.
- Dan Roth and Wen-tau Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.
- Mark Steedman. 2001. *The Syntactic Process*. Bradford Books. MIT Press.
- Alfred Tarski. 1951. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley.
- Jan van Eijck and Hans Kamp. 2011. Discourse representation in context. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language, Second Edition*, pages 181–252. Elsevier.
- Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, Massachusetts Institute of Technology, Feb. MIT AI Technical Report 235.

# Hybrid Models for Lexical Acquisition of Correlated Styles

**Julian Brooke**

Department of Computer Science  
University of Toronto  
jbrooke@cs.toronto.edu

**Graeme Hirst**

Department of Computer Science  
University of Toronto  
gh@cs.toronto.edu

## Abstract

Automated lexicon acquisition from corpora represents one way that large datasets can be leveraged to provide resources for a variety of NLP tasks. Our work applies techniques popularized in sentiment lexicon acquisition and topic modeling to the broader task of creating a stylistic lexicon. A novel aspect of our approach is a focus on multiple related styles, first extracting initial independent estimates of style based on co-occurrence with seeds in a large corpus, and then refining those estimates based on the relationship between styles. We compare various promising implementation options, including vector space, Bayesian, and graph-based representations, and conclude that a hybrid approach is indeed warranted.

## 1 Introduction

Though lexical resources are useful for many NLP tasks, manual lexicon creation is often onerous, particularly for aspects of language for where full coverage requires hundred of thousands of annotations. This work deals with one such aspect which we refer to as stylistic variation. This should not be understood in a purely aesthetic sense, but as reflecting various high-level aspects of the text, including genre and social identity. Some tasks relevant to style so defined include genre classification (Kessler et al., 1997), author profiling (Rosenthal and McKeown, 2011), social relationship classification (Peterson et al., 2011), sentiment analysis (Wilson et al., 2005), readability classification (Collins-Thompson and Callan, 2005), and text generation (Hovy, 1990). Following the classic work of Biber (1988), computational modeling of style has often focused on textual statistics and the frequency of function words and syntac-

tic categories. There are, of course, manually-constructed lists which capture some aspects of style, for instance resources related to psycholinguistics (Coltheart, 1980), but these are necessarily limited in scope. Our interest is in providing broad lexical coverage, potentially in any language. Here, we will show that style is particularly amenable to corpus-based automated lexical acquisition.

Our approach to this problem is grounded in methods popularized for polarity lexicon creation (Turney and Littman, 2003), but we take a more holistic view than is typical, simultaneously tackling the acquisition of several styles in a single model. Not only is this theoretically warranted, due to the correlation effects resulting from the oral/literate spectrum of register, but we also show it can offer practical gains: our hybrid models first derive initial estimates of each style from a large social media corpus, and then refine these estimates based partially on the results from other styles. We demonstrate that various popular methods are applicable to this problem, and indeed a single method might not provide the best results for all styles. For evaluation, we use a consensus annotation, the results of which also raise interesting questions about annotation for more continuous kinds of variation.

## 2 Related Work

In English manuals of style and other prescriptivist texts (Strunk and White, 1979; Kane, 1983), writers are urged to pay attention to various aspects of lexical style, including elements such as familiarity, readability, formality, fanciness, colloquialness, specificity, concreteness, and objectivity; these stylistic categories reflect common aesthetic judgments about language, but are also inextricably linked to the conventions of register and genre. See Biber and Conrad (2009) for a discussion of the relationship between register, genre,

and style as traditionally defined in descriptive linguistics. Some researchers have posited a few fixed styles (Joos, 1961) or a small, discrete set of situational constraints which determine style and register (Halliday and Hasan, 1976); by contrast, the applied approach of Biber (1988) and theoretical framework of Leckie-Tarry (1995) offer a more continuous interpretation of register variation. In Biber’s approach, functional dimensions such as *Involved vs. Informational*, *Argumentative vs. Non-argumentative*, and *Abstract vs. Non-abstract* are derived in an unsupervised manner from a mixed-genre corpus, with the labels assigned depending on where features (a small set of known indicators of register) and genres fall on each spectrum. The theory of Leckie-Tarry posits a single main cline of register with one pole (the oral pole) reflecting a reliance on the context of the linguistic situation, and the other (the literate pole) reflecting a reliance on cultural knowledge. The more specific elements of register are represented as subclines which are strongly influenced by this main cline, creating probabilistic relationships between related dimensions.

Computational linguistics research most similar to ours has focused on classifying the lexicon in terms of individual aspects relevant to style (e.g. formality, specificity, readability, and concreteness) (Brooke et al., 2010; Pan and Hsieh, 2010; Kidwell et al., 2009; Turney et al., 2011). Of particular methodological relevance is work on the induction of polarity lexicons based on co-occurrence in large corpora (Turney and Littman, 2003; Velikovich et al., 2010), or connections in WordNet (Rao and Ravichandra, 2009; Baccianella et al., 2010); semi-supervised vector space and graph methods are common, and several of the methods we apply here are taken directly from or inspired by work in this area.

### 3 Word annotation

In this study, we consider six styles—colloquial, literary, concrete, abstract, subjective, and objective—which are clearly represented in the lexicon, which are mentioned often in the relevant English linguistics literature, and which have strong positive and negative correlations with other styles in the group. Many (but not all) of these correlations are related to the oral/literate distinction. Our definition of each style (adapted from our annotation guidelines) is given below.

**Colloquial** Words which are used primarily in very informal contexts, for instance slang words and internet abbreviations.

**Literary** Words which you would expect to see primarily in literature; these words often feel old-fashioned or flowery.

**Concrete** Words which refer to events, objects, or properties of objects in the physical world that you would be able to see, hear, smell, or touch.

**Abstract** Words which refer to something that requires major psychological or cultural knowledge to grasp; complex ideas which can’t purely be defined in physical terms.

**Subjective** Words which are strongly emotional or reflect a personal opinion.

**Objective** Words which are emotionally distant, explicitly avoiding any personal opinion, instead projecting a sense of disinterested authority.

Our method and evaluation relies on having a set of seed words for each style. The words used in this study were originally collected from various sources by the authors; we included words that we considered clear members of a particular stylistic category—though they might also belong to other categories—with little or no ambiguity with respect to that style. Colloquial seeds consist of English slang terms and acronyms, e.g. *cuz*, *gig*, *asshole*, *lol*. The literary seeds were primarily drawn from web sites which explain difficult language in texts such as the Bible and *Lord of the Rings*; examples include *behold*, *resplendent*, *amiss*, and *thine*. The concrete seeds all denote physical objects and actions, e.g. *shove* and *lamppost*, while the abstract seeds all involve nontrivial concepts *patriotism* and *nonchalant*. For our subjective seeds, we used an edited list of strongly positive and negative terms from a manually-constructed sentiment lexicon (Taboada et al., 2011), e.g. *gorgeous* and *depraved*, and for our objective set we selected words from sets of near-synonyms where one was clearly an emotionally-distant, formal alternative, e.g. *residence* (for *home*) or *occupied* (for *busy*). We filtered initial lists to 150 of each type (900 in total), removing words which did not appear in the corpus or which occurred in multiple lists.

Relying on a single annotator, however, is problematic, and a more serious issue with our original



Table 1: Fleiss’s kappa for 5-way annotation, by style.

Style	Kappa
Literary	0.61
Abstract	0.37
Objective	0.55
Colloquial	0.85
Concrete	0.67
Subjective	0.63
Average	0.61

seed sets is that many of the seeds belong on multiple lists, reflecting the fact that stylistic correlations occur at the lexical level. This interferes with evaluation, since we need to be fairly certain not only which seeds are in a category, but which are not. Therefore, we carried out a full annotation study with 5 annotators, asking each annotator to tag all 900 words for each of the 6 styles according to guidelines we prepared. One of the authors was included as an annotator (this annotation was carried out prior to all the others), but the other four were unfamiliar with the project; all were native English speakers with at least an undergraduate degree, and all reported reading a variety of text genres for work and/or pleasure. We provided written guidelines explaining each style in detail, and asked annotators to make judgments based on what they felt to be the most common sense. Communication among annotators was restricted during the process, but we allowed access to other resources (e.g. the internet) and answered general questions about the guidelines that came up during the process. A few annotators had obviously skewed numbers for certain styles relative to other annotators due to misinterpretation of the guidelines, and we provided non-specific feedback for revision in these cases. The Fleiss’s kappa (Fleiss, 1971) values for our 5-way annotation study are presented in Table 1.<sup>1</sup>

The kappa values in Table 1 indicate agreement well above chance, but several of the dimensions (and the average) are below the 0.67 standard for reliable annotation (Artstein and Poesio, 2008), and only one (colloquial) reaches the higher 0.8 standard. This suggests that there is a sizable subjective aspect to these judgments and we should be somewhat skeptical of the judgment

<sup>1</sup>The annotations and our guidelines are available at [http://cs.toronto.edu/~jbrooke/style\\_annotations.zip](http://cs.toronto.edu/~jbrooke/style_annotations.zip).

of any particular annotator. However, we had forced our annotators to make a boolean choice for each style, which may be somewhat inappropriate for somewhat non-discrete phenomenon like style. Taboada et al. (2011), when validating their fine-grained manual polarity lexicon (which included annotation of both polarity and strength), demonstrated that Mechanical Turk worker disagreement on a boolean task seemed to correspond fairly well to ranges on a scale: there was agreement at the extremes of polarity, but increasing disagreement towards the middle.

With this in mind, we used our initial annotations to create a new annotation task for two of our external annotators: the goal was to investigate whether annotators can identify relative differences in degree suggested by either agreement or disagreement with their choices by other annotators. First, we extracted minority opinions, defined here as word/style combinations where the annotator agreed with exactly one other annotator and disagreed with the three others, and consensus opinions, defined as those where all the annotators agreed. We randomly paired each minority opinion word/style with a consensus opinion; for both opinions, the annotator in question had made the same judgment (both yes, or both no), but some of the other annotators had made different choices. We then asked our annotators (who were unaware of the exact nature of the experiment) to pick, among two words they had tagged the same in the first round, the word which had ‘more’ of the relevant stylistic quality.

In the negative case (where the annotator had originally marked both as not having the style), the results are stark: in 97% of the cases, the annotator picked the minority opinion (i.e. the word which some other annotators had marked yes), suggesting that the annotator could identify the stylistic tendencies of the (mixed-agreement) word, but had nonetheless excluded it, probably because there were much clearer examples of this style and other styles which could be more clearly applied to the word. In the positive case, the annotators preferred the word with group consensus 82.7% of the time, which is indeed the pattern we would predict if the minority opinion is less extreme; the positive case is more subtle than the negative case, where many of the words used for comparison very clearly do not belong to the relevant style. These results are consistent with the

Table 2: Number of seeds, by style.

Style	Positive	Negative
Literary	132	660
Abstract	107	599
Objective	245	495
Colloquial	163	684
Concrete	190	572
Subjective	258	487

idea that disagreement is a rough indicator of degree, and that not all disagreement should be dismissed as noise or some other failure of annotation. Of course, this also indicates that relative or continuous (e.g. Likert scale) judgments might be preferable to boolean ones, but in this case boolean annotation is far more practical, and indeed desirable for both model creation and evaluation.

For our final seed set, our positive annotations include all word/style combinations where a majority of annotators marked yes, whereas our negative annotations include only terms where there was complete consensus; words where only 1 or 2 annotators marked yes were removed from consideration as seeds (for that particular style). The summary of the counts for main seed set are presented in Table 2.

## 4 Methods

Our method for stylistic lexicon acquisition breaks down into three steps. The first is to apply one of several methods which leverages co-occurrence in a large corpus to derive, for each word, a raw score for each style. We then take that raw score and normalize it; the resulting number can be used directly to compare words relevant to a style. Finally, we consider the vector formed by these normalized style scores, and apply other methods which further refine this vector, implicitly taking into account the correlations among styles. The elements of the refined vector correspond to the degree of each style, so if we apply this method for all words in our vocabulary we create a full-coverage lexicon.

### 4.1 Corpus analysis

For all the methods in this section, we use the same corpus, the ICWSM Spinn3r 2009 dataset (Burton et al., 2009), which has been used successfully in earlier work (Brooke et al., 2010). Social media corpora are particularly appropriate for research

on style, since they contain a variety of registers. Here, we include all 2.46 million texts in the Tier 1 portion which contained at least 100 word types. Hapax legomena were excluded, since they could not possibly offer any co-occurrence information, but otherwise we did not filter or lemmatize words: our full vocabulary is 1.95 million words.

Our simplest method uses pointwise mutual information (PMI) (Church and Hanks, 1990), a popular metric for measuring the association between words. Since standard PMI has a lower bound of  $-\infty$  when the joint probability is 0 (a common occurrence since many of our words are relatively rare), we actually use a normalized version, NPMI, which has an upper bound of 1 and a lower bound of  $-1$ .

$$NPMI(x,y) = \left( \log \frac{p(x,y)}{p(x)p(y)} \right) \left( \frac{1}{\log p(x,y)} \right)$$

Following earlier work (Brooke et al., 2010), here and elsewhere we do not use the term frequency within a document (which is less relevant to style). Instead the probabilities are calculated using the number of documents where the word or words appear divided by the total number of documents. The raw score  $r_{ij}$  for style  $i$  of word  $w_j$  is simply the sum of its NPMI with the associated set of seeds  $S_i$ :

$$r_{ij} = \sum_{s \in S_i} NPMI(w_j, s)$$

Our second method, LSA, was applied to formality by Brooke et al. (2010) and concreteness by Turney et al. (2011). We begin by converting our corpus into a binary word-document matrix, and carry out latent semantic analysis (Landauer and Dumais, 1997), which includes a singular value decomposition of the matrix and dimensionality reduction to  $k$  dimensions. Assuming  $\mathbf{v}_w$  denotes the resulting  $k$ -dimensional vector for word  $w$ , we calculate  $r_{ij}$  as:

$$r_{ij} = \sum_{s \in S_i} \cos(\theta(\mathbf{v}_{w_j}, \mathbf{v}_s))$$

Our third method, using latent Dirichlet allocation (Blei et al., 2003), is more novel for lexical acquisition, and we address the specifics of this method in more detail in other work (Brooke and Hirst, 2013). Briefly, LDA is a Bayesian topic model which assumes that texts are generated via a

distribution of topics for each text ( $\theta$ ), and a distribution of words for each topic ( $\beta$ ); given a corpus, appropriate values for  $\theta$  and  $\beta$  are derived using inference, in this case variational Bayes inference using the original implementation provided by Blei et al. (2003). Our method works by seeding each of six topics in an LDA model (corresponding to our six styles) by dividing the entire initial probability mass among the seeds and running two iterations of the model, which distributes some of the probability mass to co-occurring words. In our previous work, we found further iterations had no benefit and even slightly degraded the model. For the LDA method,  $r_{ij}$  corresponds directly to  $\beta_{ij}$  of the resulting model which is just the probability of topic (style)  $i$  generating  $w_j$ .

## 4.2 Normalization

The raw numbers derived from corpus analysis methods discussed above cannot be used directly as indicators of style: the frequencies of both the seeds and the words being predicted have significant effect on the relative and absolute magnitudes of each style for all our methods, and performance using just these numbers is near chance. However, in two steps we can normalize these numbers to a form where the magnitude does directly reflect degree of a style. Again,  $r_{ij}$  refers to the raw score for style  $i$  and word  $j$  from some corpus analysis method. First, we take steps to ensure that  $r_{ij}$  is nonnegative. For LDA this is unnecessary (since  $r_{ij}$  is based on a probability distribution), but for NPMI and LSA it is needed, since both involve summing over items which vary between  $-1$  and  $1$ . We can ensure that these are positive by adding a constant equal to the number of seeds. Next, we convert the result to a style ‘distribution’ for each word:

$$r'_{ij} = \frac{r_{ij} + |S_i|}{\sum_{k=1}^6 r_{kj} + |S_k|}$$

The result is still not useful, since frequency (and count) of seeds clearly still has an effect. To focus on the differences between words, we subtract the means for each style and divide by the standard deviation

$$b_{ij} = \frac{r'_{ij} - \bar{r}'_i}{\sigma_{r'_i}}$$

to reach  $b_{ij}$ , the base for the ‘style space’ methods in the next subsection.

## 4.3 Style Vector Optimization

Given a vector that represents the styles for a given word, we wish to refine the vector to improve performance on relative judgments for individual styles. Here, we test two options: the first transforms the stylistic vectors into  $k$ -Nearest Neighbor (kNN) graphs, where we can apply label propagation. The second option treats the vector as a set of features for supervised linear regression, one for each style, using a specialized loss function. Both methods rely on having a style vector representation of not only our target words, but also our seed (training) words. For LSA and NPMI, we used leave-one-out crossvalidation to create these vectors; for LDA, however, it was impractical to do a full run of the model for each word, and so we used 10-fold crossvalidation instead.

A vector-space representation offers a number of obvious similarity functions for building a  $k$ NN graph: we test two here, inverse Euclidean distance (L2) and cosine similarity (cos). A more difficult problem is the choice of  $k$  (for  $k$ NN  $k$ ): here, we estimate a good  $k$  from the training set. Since the training set and dimensionality of the data is (now) fairly small, we simply test on all possible intervals of 5, and choose the best (often near 50, though we saw values as low as 10 and as high as 90) using our pairwise evaluation (see Section 5.1). Since our label propagation method works independently for each style, we can choose a different  $k$  for each.

For label propagation, we use the simple one-step propagation function from Kang et al. (2006). Here,  $K$  is our similarity function (which returns zero if seed  $s$  is not one of the  $k$  nearest neighbors), and  $z_{ij}$  is the resulting confidence score, which we use as our new estimate for the style:

$$z_{ij} = \sum_{w_s \in S_i} K(w_j, w_s)$$

Obviously, the main work here is done by the similarity function, which implicitly includes information from other stylistic dimensions by preferring words which are close not just on the relevant dimension, but in the stylistic space as a whole. There are of course more sophisticated, multi-step approaches to label propagation, e.g. the one used by Rao and Ravichandran (2009), but a single-step approach has clear advantages in light of our large vocabulary and dense graph; we leave exploration of whether unlabeled words can help further to fu-

ture work. We did test the one-step correlated label propagation method proposed by Kang et al. but found it was ineffective, probably because it increases the effects of correlation, which is actually counter to our needs.

The information provided by label propagation is distinct enough that it can be successfully combined with the original (base) vector. As with  $k$  for  $k$ NN, we estimated a good weighting for this combination using the training data, testing at 0.01 intervals. Since we noted some interdependence, we combined this step with the selection of ( $k$ NN)  $k$ . Again, this ratio can be different for each style.

Our second vector optimization technique is an adaption of supervised linear regression. Linear regression usually involves minimizing squared distance of the output of the model from the training set, assuming there are known values of expected output. In this case, however, we don't have reliable values for specific degrees of a style. We proceed by replacing the least-squared loss function with a loss function based on our evaluation metric (see Section 5.1):

$$L(\theta) = \sum_{w_j \in S_{i,p}} \sum_{w_m \in S_{i,n}} I(h_\theta(b_{ij}) < h_\theta(b_{im}))$$

Here,  $S_{i,p}$  and  $S_{i,n}$  refer to the positive and negative examples of style  $i$ , respectively,  $h_\theta$  is the linear regression function, and  $I$  is an indicator function equal to 1 if the statement is true, and 0 otherwise.

Using such a loss function discourages standard approaches to linear regression, but in this context (a small feature space and training set), it is reasonably practical to search the space exhaustively for weights which provide a (near-)optimal result (on the training data).<sup>2</sup> Starting with full weight (1) on the feature corresponding to the dimension being derived and 0 on all others, we search the range  $-1$  to  $1$  at 0.001 intervals for the other dimensions, proceeding in order based on the greatest difference across positive and negative examples of each style. We found that one such iteration across each element of the vector was sufficient, resulting in a stable model. This method can be applied on the initial vector, or on a vector that has already been refined by some other method, i.e. the output of label propagation.

<sup>2</sup>At the suggestion of a reviewer, we also tried applying SVMrank to this regression; it was much faster but performance was worse.

## 5 Evaluation

### 5.1 Setup

Our evaluation is based on the pairwise comparison of words which are known (from our annotation) to differ relevant to a certain style. Accuracy for a test set  $S_i$  (of a style  $i$ ) is defined as the number of instances where the expected inequality exists between a pair of opposing words, divided by the total number of such pairings:

$$Accuracy(S_i) = \frac{\sum_{w_j \in S_{i,p}} \sum_{w_m \in S_{i,n}} I(z_{ij} > z_{im})}{|S_{i,p}| \cdot |S_{i,n}|}$$

Here  $z$  can refer to any of the metrics for style discussed in the previous section. The major advantage of this definition of accuracy is that it does not require an arbitrary cutoff point, but 100% accuracy nonetheless indicates that the two sets are perfectly separable. Also, it does not assume anything about the degree of difference between two words, e.g. that more is better, since for any given pair of words we cannot be certain what an ideal difference would be.

We evaluate using 3-fold crossvalidation, using the original 150-per-style annotation of our 900 words for the purposes of stratifying the data, which allows for balanced sets of 600 for training and 300 for testing. All seeding, training, and evaluation use the majority annotation of the 5 annotators, discussed in Section 3. Since the initial splits add a significant random factor, all results here are averaged over 5 runs, with the same 5 runs (i.e. same splits) used for all evaluated conditions.

### 5.2 Comparison of models

Table 3 shows a comparison of the performance of various models, organized by the method of corpus analysis. First, we note that most of these numbers are quite high, almost all are above 80% and most are above 90%. It is worth mentioning that if only direct opposites are considered (e.g. colloquial versus literary, concrete versus abstract), most dimensions reach results above 99%; our multi-style evaluation here offers a more realistic view. Among individual styles, colloquial words seem the most distinct, which is consistent with the results of human annotation. Acquisition of subjectivity, on the other hand, is strikingly more difficult than the other styles.

Based only on average accuracy, we could conclude that  $LSA > LDA > NPMI$  with respect

Model	By Style						Average
	Lit.	Abs.	Obj.	Coll.	Conc.	Subj.	
guessing baseline	50.0	50.0	50.0	50.0	50.0	50.0	50.0
<b>NPMI</b>							
base (Normalized)	68.4	91.2	94.4	95.6	73.4	77.1	83.0
LP-cos	90.1	91.5	95.1	94.4	90.0	80.0	90.2
LP-L2	88.2	88.9	94.1	94.1	89.4	76.6	88.5
base+LP-cos	90.2	92.8	95.6	96.0	90.6	80.9	91.0
base, LR	89.8	93.6	94.2	96.5	85.5	79.7	89.9
base+LP-cos, LR	90.2	93.6	95.5	95.9	90.5	81.0	91.1
<b>LDA</b>							
base	67.3	93.3	96.5	96.2	93.2	83.5	88.3
LP-cos	86.0	92.9	96.0	93.6	94.8	86.5	91.6
LP-L2	78.1	91.1	95.0	92.5	94.2	83.2	89.0
base+LP-cos	86.4	93.5	96.6	96.3	95.5	86.7	92.5
base, LR	84.3	93.9	96.5	96.4	94.7	85.7	91.8
base+LP-cos, LR	87.2	93.9	96.5	96.3	<b>95.8</b>	<b>87.0</b>	92.8
<b>LSA</b>							
k=20, base	89.1	93.5	95.6	94.4	90.8	76.0	89.9
k=500, base	91.2	93.7	96.5	96.5	93.7	83.5	92.6
k=500, LP-cos	92.4	91.7	96.0	96.8	94.3	85.2	92.8
k=500, LP-L2	92.1	92.1	96.5	96.5	94.3	85.0	92.8
k=500, base+LP-cos	92.5	93.6	96.8	97.5	94.8	85.9	93.5
k=500, base, LR	<b>92.7</b>	<b>94.0</b>	<b>97.2</b>	97.2	94.9	86.5	<b>93.7</b>
k=500, base+LP-cos, LR	<b>92.7</b>	93.8	97.0	<b>97.7</b>	94.9	86.4	<b>93.7</b>

Table 3: Model performance in lexical induction of seeds, % pairwise accuracy. LP = label propagation, cos = cosine similarity, L2 = inverse Euclidean distance, LR = linear regression. Bold is best in column.

to extracting relevant stylistic information from the corpus. That NPMI is the worst performing method is not surprising, since it relies only on direct co-occurrence between seeds and test words, and is not able to take advantage of larger patterns in the data; we would expect similar results for other simple relatedness measures. Though LSA is better overall, the distinction between LSA and LDA is more subtle, since in fact LDA is the higher performing model for two of the six styles, and its poorer overall performance can be attributed to a rather dismal showing for literary words, worse than NPMI. This is interesting because subjective and concrete words, where LDA does well, are the most common in the corpus, whereas literary words are consistently the least common. We posit, based on this and our earlier research focused on the LDA method, that successful low-dimensional seeded LDA requires styles (topics) that are reasonably well-represented in the corpus; when that condition is met, LDA will likely do better than LSA because it will

distinguish rather than collapse correlated styles. LSA, on the other hand, is robust against the scarcity problem because it requires only that a set of words have a reasonably distinct  $k$ -dimensional profile to form a coherent style.

Based on the results in Table 3, we can conclude decisively that both of our optimization techniques are effective. The effects are particularly marked for NPMI, but is reasonably consistent across all three corpus analysis techniques and the various individual styles. With regards to the similarity function in label propagation, we found that cosine similarity, a less common choice for building graphs, was generally as good as, and often better than, Euclidean distance. The vector resulting from label propagation also consistently benefited from being combined with the base vector, the result being better than either alone. It is not entirely clear which of the two optimization methods is to be preferred (their effects seem roughly similar), though linear regression seems to have edge when using LSA. Combining the two methods seems a

good strategy, particularly for LDA.

The LSA results presented here mostly use  $k = 500$ , a fairly standard choice. However, we tested other values, in particular extremely low values ( $k = 20$ ) to see if we could confirm our supposition (Brooke et al., 2010) that much stylistic information is contained with the first few dimensions of LSA. Our results suggest that the basic supposition is valid, since the difference between the two conditions for most dimensions is not large, but the identification of subjectivity (not considered by Brooke et al. 2010) does seem to benefit greatly from a higher-dimensional vector.

## 6 Qualitative analysis

To investigate further the successes and failures of our method, we carried out two qualitative examinations of the output of our model. First, we looked at those words within our annotated set of words which consistently caused the most errors across the various splits and runs. Second, we ran a high-performing LSA model built from the entire seed set on a subset of our vocabulary (we excluded words of document frequency less than 100), creating lexicons for each style; we manually inspected non-seed words that were ranked highest on each dimension.

The clearest result from the inspection of the seed output was that many of the false negatives involve words that are strong on some other dimension, typically on the other side of the oral/literate divide. For example, the most difficult-to-identify literary and abstract terms are strongly subjective (e.g. *loathe* and *obscene*), while the most difficult objective word, *translucent*, is very concrete. The most difficult concrete words are literary (*yoke*, *raiment*) or objective (*conflagration*), and the most difficult subjective words are also somewhat objective (*eminent*) or abstract (*autocratic*). Interestingly, a manual inspection of the weights for linear regression suggests that our optimization is correcting for just this kind of situation: we generally see negative weights on (what we would predict to be) positively correlated styles, and vice versa. However, in certain cases where one style has a much larger role in determining the co-occurrence pattern in the corpus, this correction may be insufficient.

Most of the false positives, by contrast, involve overextension of each category in predictable ways. For example, our highest ranking literary

words from the general vocabulary were mostly very good, but contained a few words that are obvious over-generalizations into biblical and fantasy texts, e.g. *locust* and *sorcerers*, while among the objective words there were a number of academia-relevant words that are really more abstract than objective, e.g. *coauthors* and *peer-review*. Our derived colloquial words contained many (sometimes purposeful) misspellings (*wayy*, *annnd*) which we could argue are genuinely colloquial; less clear are the many lower-case celebrity names (e.g. *miley*), but the fact that the bloggers used lower case does make them non-standard. Consistent with our qualitative results, subjective was the most problematic in the general vocabulary: though there were many good subjective words, there were a lot of other words which suggest topics that people tend to express opinions about, e.g. *sitcoms*, *entertainer*, or *flick*; movie-related words are particularly common, which might be a reflection the lexicon we took our subjective seeds from.

## 7 Conclusion

We have presented a methodology for deriving high-quality stylistic lexicons from corpora. A key aspect of our approach its hybrid nature: information is first extracted (using efficient, well-established methods) in a semi-supervised fashion from large corpora, and then refined using fully-supervised techniques. We argue that there are clear benefits in looking at multiple styles simultaneously, not only in terms of improving performance but also in taking our evaluation beyond ‘toy’ situations where we ignore the complexities and interactions among styles, drawing connections with broader insights from linguistics.

One possible criticism of our method is that we use only co-occurrence information, and not other information (e.g. word morphology) which could be relevant to particular styles in English; this option should be explored further, particularly in the optimization phase where we can easily add other features, though we stress that our ultimate goal is to derive methods that are easily extensible to more styles and more languages. We have also not considered word senses or multiword expressions, but both can and should be added to the model.

## Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

## References

- 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 *Proceedings of LREC'10*, Valletta, Malta.
- Douglas Biber and Susan Conrad. 2009. *Register, Genre, and Style*. Cambridge University Press.
- Douglas Biber. 1988. *Variation Across Speech and Writing*. Cambridge University Press.
- David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Julian Brooke and Graeme Hirst. 2013. A multi-dimensional Bayesian approach to lexical style. In *Proceedings of NAACL '13*, Atlanta.
- Julian Brooke, Tong Wang, and Graeme Hirst. 2010. Automatic acquisition of lexical formality. In *Proceedings of COLING '10*, Beijing.
- Kevin Burton, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In *Proceedings of ICWSM '09*, San Jose.
- Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.
- Kevyn Collins-Thompson and Jamie Callan. 2005. Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science Technology*, 56(13):1448–1462.
- Max Coltheart. 1980. *MRC Psycholinguistic Database User Manual: Version 1*. Birkbeck College.
- Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.
- M.A.K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.
- Eduard H. Hovy. 1990. Pragmatics and natural language generation. *Artificial Intelligence*, 43:153–197.
- Martin Joos. 1961. *The Five Clocks*. Harcourt, Brace and World, New York.
- Thomas S. Kane. 1983. *The Oxford Guide to Writing*. Oxford University Press.
- Feng Kang, Rong Jin, and Rahul Sukthankar. 2006. Correlated label propagation with application to multi-label learning. In *Proceedings of CVPR '06*, New York.
- Brett Kessler, Geoffrey Nunberg, and Hinrich Schütze. 1997. Automatic detection of text genre. In *Proceedings of ACL '97*, Madrid.
- Paul Kidwell, Guy Lebanon, and Kevyn Collins-Thompson. 2009. Statistical estimation of word acquisition with application to readability prediction. In *Proceedings of EMNLP'09*, Singapore.
- Thomas K. Landauer and Susan 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.
- Helen Leckie-Tarry. 1995. *Language and Context: A Functional Linguistic Theory of Register*. Pinter.
- Ching-Fen Pan and Shu-Kai Hsieh. 2010. Word space modeling for measuring semantic specificity in Chinese. In *Proceedings of COLING '10*, Beijing.
- Kelly Peterson, Matt Hohensee, and Fei Xia. 2011. Email formality in the workplace: A case study on the Enron corpus. In *Proceedings of ACL '11*, Portland.
- Delip Rao and Deepak Ravichandra. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of EACL '09*, Athens.
- Sara Rosenthal and Kathleen McKeown. 2011. Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In *Proceedings of ACL '11*, Portland.
- William Strunk and E.B. White. 1979. *The Elements of Style*. Macmillan, 3rd edition.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.
- Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.
- Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of EMNLP '11*, Edinburgh, United Kingdom.
- Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Proceedings of NAACL '10*, Los Angeles.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP '05*, Vancouver.

# Introducing the Prague Discourse Treebank 1.0

Lucie Poláková, Jiří Mírovský, Anna Nedoluzhko, Pavlína Jínová,  
Šárka Zikánová and Eva Hajičová

Charles University in Prague  
Faculty of Mathematics and Physics  
Institute of Formal and Applied Linguistics  
Czech Republic

{polakova|mirovsky|nedoluzko|jinova|zikanova|hajicova}@ufal.mff.cuni.cz

## Abstract

We present the Prague Discourse Treebank 1.0, a collection of Czech texts annotated for various discourse-related phenomena "beyond the sentence boundary". The treebank contains manual annotations of (1), discourse connectives, their arguments and senses, (2), textual coreference, and (3), bridging anaphora, all carried out on 50k sentences of the treebank. Contrary to most similar projects, the annotation was performed directly on top of syntactic trees (from the previous project of the Prague Dependency Treebank 2.5), benefiting thus from the linguistic information already existing on the same data. In this article, we present our theoretical background, describe the annotations in detail, and offer evaluation numbers and corpus statistics.

## 1 Introduction and Motivation

Large collections of gold standard language data are known to build an indispensable base for many NLP algorithms. Reliable morphological tagging and syntactic analysis (phrasal or dependency) are nowadays quite a standard information in language corpora released all over the world. With the gradually increasing interest in modeling discourse structure or using various discourse features<sup>1</sup> in different NLP tasks (anaphora resolution, summarization, MT), also the development of resources aimed at representing various discourse-related aspects has gained on importance. Moreover, both theoretical discourse research and NLP algorithms can benefit from a reliable **multi-dimensional** analysis of the data (Webber et al., 2003, Stede, 2004). There are already several elaborate theoretical concepts on

<sup>1</sup> The term of *discourse* in this paper is used in two meanings. The broader interpretation is roughly equal to *text* (as in *discourse structure*, *discourse features* or *discourse coherence*) whereas the narrower sense denotes semantic relations between propositions (as in *discourse relations*).

discourse coherence brought to life in real-data annotation (see Sections 1.1 and 1.2). Still, it is only in recent years that large-scale corpora with manual annotations of sentential **and** discourse level phenomena have become available. Even fewer such corpora exist that combine more types of manual discourse-level annotations.

In this paper, we present a large-scale manual annotation project for Czech in which, apart from the "standard" analysis of a sentence (morphology, syntactic trees), several discourse phenomena are marked, all over the same data: pronominal, nominal and zero<sup>2</sup> coreference, discourse connectives (henceforth DCs) and the semantic relations they express, and the associative relations of the so-called bridging anaphora.

The paper is structured as follows: In Sections 1.1 and 1.2, brief overviews of recent projects concerning discourse relations and coreference + bridging anaphora are described, respectively. In Section 2, data and tools used in Prague Discourse Treebank (PDiT) are introduced. Section 3 describes the annotation scenario and is followed by evaluation of the project in comparison with similar projects (Section 4) and basic distribution numbers (Section 5). We conclude with discussion (Section 6).

### 1.1 Corpora of Discourse Relations

The first attempts in representing discourse structure date over a decade back. One of very first and most influential projects was the RST-Treebank (Carlson et al., 2001), an annotation project over the English texts of Wall Street Journal. In accordance with the Rhetorical Structure Theory of Mann and Thompson (1988), the whole document is represented as a single tree-like structure. Wolf and Gibson (2005) propose a less con-

<sup>2</sup> Czech is a pro-drop language. The restored ellipses in the underlying sentence analysis allow us to annotate zero forms as co-referential.



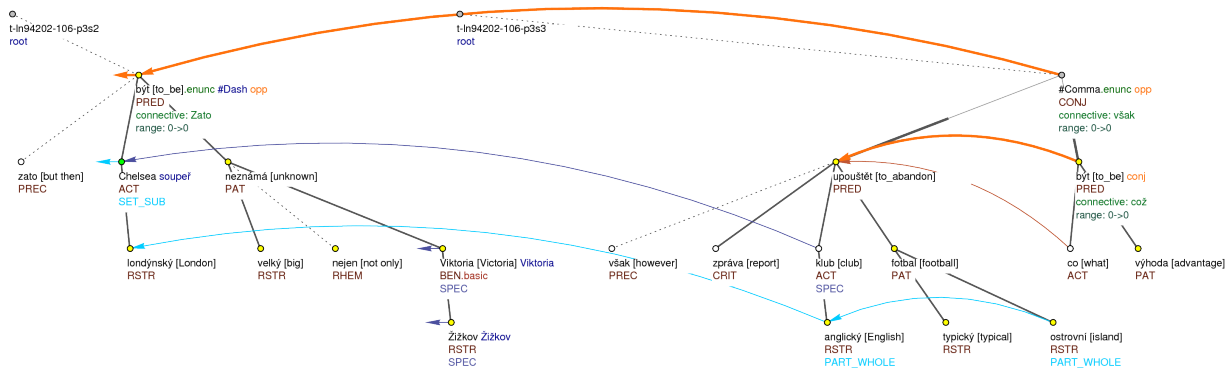


Figure 1. Annotation of two sentences. Discourse relations are represented by thick orange arrows, textual coreference by dark blue slim arrows, bridging anaphora by light blue slim arrows. Grammatical coreference (the only one in the figure is between nodes *co [what]* and *upouštět [to abandon]*) is represented by a brown slim arrow.

strained model in Discourse Graphbank by giving up the requirement of a tree-structure. These approaches are referred to as "deep discourse parsing" or modeling of global coherence (whole document = one connected structure) in contrast to the so-called "shallow discourse parsing" or local coherence modeling of the lexically grounded approaches, which are based on identification of discourse markers and relations they express. The most influential of the latter is the Penn Discourse Treebank (for English, PDTB, Prasad et al., 2008) with several subsequent similarly aimed corpora for different languages, the project presented here being one of them.

Resources manually annotated for (some type of) discourse phenomena are already available or work-in-progress for various languages, including Chinese (Zhou and Xue, 2012), Arabic (Al-Saif and Markert, 2010), Turkish (Zeyrek et al., 2010), Hindi (Oza et al., 2009), French (Afantenos et al., 2012, Danlos et al., 2012), German (Stede, 2004, Gastel et al., 2011) and others. Additionally, the relevance of the PDTB annotation concept was further tested on specific domains, e.g. on spoken dialogs (Italian, Tonelli et al., 2010) and on biomedical texts (English, Prasad et al., 2011).

## 1.2 Corpora of Coreference and Bridging Relations

There is a number of different large-scale annotated corpora for coreference and anaphoric relations. The largest annotated corpora for English include MUC (Hirschman and Chinchor, 1997), ACE (Doddington et al., 2004), OntoNotes (Pradhan et al., 2007), GNOME (Poesio, 2004), ARRAU (Poesio and Artstein, 2008). The coreference annotations for other languages than English are more limited. The most well-known corpora including anaphoric information are

AnCorra (Recasens and Martí, 2009) for Spanish and Catalan, VENEX (Poesio et al., 2004a) for spoken and written Italian, the Italian Live Memories Corpus (Rodríguez et al., 2010), TüBA-D/Z (Hinrichs et al., 2004) and Postdam Commentary Corpus (Stede, 2004, Krasavina and Chiarcos, 2007) for German, and some others.

Early work on bridging relations dates back to the mid-70s. Clark (1975) documents several ways in which an inference is needed to understand the meaning intended by the speaker. Clark names several types of bridging relations such as set-membership, part-whole, roles, reasons and consequences. Bridging relations have been later investigated by Poesio et al. (1997, 2004b). The annotation of bridging relations in different projects includes different types of relations. In the GNOME corpus (Poesio, 2004), such bridging relations as set-membership, subset, and part-whole are annotated. The Copenhagen Dependency Treebank (Korzen and Buch-Kromann, 2011) has a very detailed annotation scheme based on general semantic roles. Another way to capture bridging relations is to define them vaguely, e.g. as a reference which is made to a subpart of an object that has already been mentioned in the discourse (Hendrickx et al., 2011) or to mark as bridging all non-coreferent anaphoric references. The last approach was used in Hou et al. (2013), providing a reasonably sized and reliably annotated corpus for English.

To our knowledge, there are only few corpus projects portraying phenomena "beyond the sentence boundary" that gather different types of textual information, or, in other words, offer some kind of multi-dimensional discourse annotation. The texts of Wall Street Journal have undergone various annotations but they arose within different projects and frameworks – rhet-

TEMPORAL	CONTINGENCY	CONTRAST	EXPANSION
synchronous	reason – result	confrontation	conjunction
asynchronous	<i>pragmatic reason – result</i>	opposition	exemplification
	condition	<i>pragmatic contrast</i>	specification
	<i>pragmatic condition</i>	restrictive opposition	equivalence
	explication	concession	generalization
	purpose	correction	conjunctive alternative
		gradation	disjunctive alternative

Table 1: Distribution of discourse types in the data

orical structure analysis in RST-Treebank (385 WSJ articles), Discourse Graphbank (135 texts from AP Newswire and WSJ), Penn Discourse Treebank 2.0 (2,159 WSJ articles), OntoNotes (a substantial portion of the WSJ-Penn Treebank annotated for coreference) etc. A multi-dimensional analysis within a single project was conducted for French in AnnoDis (Afantenos et al. 2012, an intersection of all annotations on 13 articles), for German in the Potsdam Commentary Corpus (Stede, 2004, 170 texts), and lately in TüBa-D/Z (Gastel et al., 2011, 919 sentences in 31 articles). These projects include inter alia some particular version of a "global" discourse analysis, annotation of connectives and their senses, and coreference annotation.

## 2 Data and Tools

As the base data for the annotation, we used the Prague Dependency Treebank 2.5 (PDT, Bejček et al., 2012), which is an update of the Prague Dependency Treebank 2.0 (Hajič et al., 2006). It is a treebank of almost 50 thousand sentences of Czech newspaper texts, annotated manually on three levels of annotation: morphological, analytical and tectogrammatical. The annotation of a sentence at the highest, tectogrammatical layer captures the deep syntax and the information structure of a sentence and is represented by a dependency tree.

For the annotation of discourse relations, textual coreference and bridging anaphora, we used several extensions to a highly customizable tree editor TrEd (Pajas and Štěpánek, 2008). Technically, each of the annotated relations is represented as an arrow connecting two tectogrammatical nodes. The two nodes represent the two arguments of the relation, i.e. typically the subtrees of the nodes. All information about the relation is kept in a set of dedicated attributes at the initial node of the relation, containing a unique identifier of the target node of the relation, type of the relation, and other pieces of information (depending on the relation, e.g. a connective for the

discourse relation). The relation is depicted as a curved arrow between the nodes, see Figure 1. For details on the annotation tool for discourse, see Mírovský et al. (2010a), for details on the annotation tool for textual coreference and bridging anaphora, see Mírovský et al. (2010b).

## 3 Annotation

The following subsections 3.1 and 3.2 describe the annotation principles for the two subprojects in PDiT, the annotation of discourse relations and the annotation of textual coreference and bridging anaphora. Detailed descriptions of the annotation guidelines can be found in annotation manuals (Poláková et al., 2012a, Nedoluzhko et al., 2011). Figure 1 shows the annotation of two sentences in Example 1 in all these aspects.

(1) *Zato londýnská Chelsea je velkou neznámou nejen pro Viktorii Žižkov. Podle zpráv však anglický klub upouští od typického ostrovního fotbalu, což by mohlo být výhodou.*

*But then London Chelsea is a big unknown not only for Victoria Žižkov. According to reports, however, the English club abandons the typical island football, which could be an advantage.*

### 3.1 Discourse

Annotating discourse relations in PDiT is inspired by the PDTB lexical approach of connective identification (Prasad et al., 2008) but it also takes advantage of the Prague tradition of dependency treebanking. This means in practice that some discourse information (intra-sentential) could have been extracted from the previous rich annotation of syntax, with only minor enhancements (Jínová et al., 2012b). In the first release of PDiT, we only focused on discourse relations indicated by overly present (explicit) discourse connectives, i.e. expressions like *but*, *however*, *as a result*, *even though* etc.<sup>3</sup> Every DC is thought of as a discourse-level predicate that

<sup>3</sup> Some remarks on annotation of the implicit DCs and of the so-called alternative lexicalizations of connectives (AltLex) are added in the discussion in Section 6.

takes two discourse units as its arguments. Only discourse relations connecting clausal arguments (with a predicate verb), i.e. not those between nominalizations or deictic expressions were annotated in version 1.0. Additionally, the Prague discourse annotation includes marking of list structures (as a separate type of discourse structure) and marking of some smaller text phenomena: article headings, figure captions, non-coherent texts like collections of news etc.

The annotation of discourse relations consisted of two phases, first being manual and the subsequent including automatic extraction of relevant syntactic features. For the manual part, the annotators had at their disposal both plain text and the tree structures, the annotation itself was carried out on syntactic (tectogrammatical) dependency trees, as we did not want to lose connection with and information from the analyses of previous levels. Intra-sentential discourse relations, i.e. those that had already been captured within the syntactic (tectogrammatical) analysis, were only to be newly annotated if their discourse semantics differed from the tectogrammatical interpretation (Jinová et al., 2012b), otherwise they were automatically extracted and mapped onto the discourse annotation.

#### Automatic Extraction of Syntactic Features

An automatic procedure was designed to extract discourse-relevant features from the syntactic level of description, i.e. the intra-sentential discourse relations. As mentioned earlier, the tectogrammatical tree structures offer some types of information that can be transferred to the discourse-level annotation. In general, this concerns subordinate syntactic relations between clauses with labels like causality, conditionality, temporality, concession etc.; and coordinate syntactic relations between clauses of one sentence with selected coordinative labels like conjunction, disjunction, opposition or contrast, confrontation etc. These relations were semi-automatically mapped onto the discourse annotation. (Jinová et al., 2012b).

#### Semantic labels

The Prague discourse label set was inspired by the tectogrammatical functors (Mikulová et al., 2005) and also by Penn sense tag hierarchy (Miltsakaki et al., 2008). Table 1 shows the discourse-semantic label set used for PDiT 1.0. The four main semantic classes, Temporal, Contingency, Contrast (Comparison) and Expansion are identical to those in PDTB but the hierarchy it-

self is only two-level. The third level is captured by the direction of the discourse arrow. The annotators, unlike in the Penn approach, were not allowed to only assign the major class, they always had to decide for a single relation within one of the classes.<sup>4</sup> Within these four classes, the types of the relations partly differ from the Penn types and go closer to Prague tectogrammatical functors and/or are a matter of language-specific distinctions. Compared to the PDTB label set, we added the categories of *purpose* and *explication* in the Contingency group and *restrictive opposition* and *gradation* to the Contrast group. In the PDTB, four pragmatic meanings are distinguished and annotated: *pragmatic cause*, *condition*, *contrast* and *concession*. In the Prague scenario, three pragmatic senses were annotated, pragmatic concession and pragmatic contrast joined to one group, for the lack of reliable distinctive features.<sup>5</sup>

#### Post-annotation checks and fixes

After the manual annotation of discourse relations was finished, some checks turned up to be necessary, especially for relations whose nature revealed to be more complicated in real data than we had expected on the basis of linguistic handbooks. After having collected all examples of these relations (namely *specification*, *explication*, *generalization*, *exemplification* and *equivalence*) in our data and established more complex definitions of their nature, annotation of these relations was manually unified in the whole data. Also some DCs required unification via post-annotation. Additionally, the part of the data which was annotated first was fully re-annotated at the end since we expected it might have suffered from initial inexperience of the annotators.

Results of the automatic extraction were checked randomly on several hundreds of examples. All discrepancies found were integrated in an automatic script (treatment of multiple DCs, multiple coordinations etc.). Only two situations required manual checks and fixes: i) Due to a complicated situation in a tree, the automatic extraction failed in 23 cases of DC identification (opposed to 10,482 cases with correct identification). ii) Solely manual treatment was necessary for constructions with a discourse-relevant clause dependent on a complex predicate structure with

<sup>4</sup> In special cases, they had the option to assign an additional secondary relation.

<sup>5</sup> It may be that different text types require slightly different sets of semantic labels. For instance, some discourse projects use a more fine-grained set of pragmatic senses (e.g. for spoken dialogs).

an infinitive or a noun phrase. In such cases only semantics allowed to distinguish if the clause is related to the whole structure or only to the infinitive or noun phrase.<sup>6</sup>

### 3.2 Coreference and Bridging Relations

In PDiT 1.0, two types of coreference (grammatical and textual) and six types of bridging relations are marked. The **grammatical coreference** typically occurs within a single sentence, the antecedent being able to be derived on the basis of grammatical rules of a given language (Czech). It includes relative pronouns, verbs of control, reflexive pronouns, reciprocity and verbal complements (Mikulová et al., 2005). **Textual coreference** marks coreferential relations between language expressions referring to the same discourse entity when the reference is not expressed by grammatical means alone, but also via context. Anaphoric (occasionally cataphoric) relations are expressed by various linguistic means (pronouns, synonyms, generalizing nouns etc.). Textual coreference has been annotated in two time periods. First, the so-called pronominal textual coreference was manually annotated. It was restricted to cases in which a demonstrative *this* or an anaphoric pronoun of the 3rd person, also in its zero form, are used (Kučová and Hajičová, 2004). Afterwards, the annotation of textual coreference was extended to cases where the anaphoric expression is represented by other means such as full noun phrases, adverbs (*there, then* etc.) and some types of numerals and pronouns left out during the first stage (Nedoluzhko et al., 2013).

The textual coreference is further classified into two types – coreference of noun phrases with specific (type SPEC) or generic (type GEN) reference. Compare examples (2) and (3):

(2) *Mary* and John went together to Israel, but *Mary* [type SPEC] had to return because of the illness.

(3) *Dogs* bark. This is the way how *they* [type GEN] express their emotions.

Discourse deixis (reference to a non-nominal antecedent) is annotated as a textual coreference link when referring to a clause or a sentence. If a noun phrase endophorically refers to a discourse segment that is larger than one sentence or it is understood by inferencing from a broader context, the antecedent is not specified.<sup>7</sup>

A specifically marked link for **exophora** denotes that the referent is "out" of the co-text, it is known only from the actual situation. In the same way as for segments, the new nominal and adverbial links were added.

For the **bridging relations**, the following types are distinguished: part-of relation (*room - ceiling*), set – subset (*students – some students*) and FUNCT (*trainer – football team*) traditional relations, CONTRAST for coherence relevant discourse opposites (e.g. *this year – next year*), ANAF for explicitly anaphoric relations without coreference (*second world war – at that time*) and the further underspecified group REST, which is mainly used to capture such types of bridging relations as location – inhabitants or event – argument. A more detailed description of the types can be found in Nedoluzhko and Mirovský (2011).

#### Automatic Preannotation

For the textual coreference, only a limited preannotation was carried out: We used a list of pairs of words that with a high probability form a coreferential pair in texts. Most of the pairs in the list consist of a noun and a derived adjective, which are different in Czech, e.g. Praha – pražský (in English: Prague – Prague, like in the sentence: *He arrived in Prague and found the Prague atmosphere quite casual*). The rest of the list is formed by pairs consisting of an abbreviation and its one-word expansion, e.g. ČR – Česko (similarly in English: USA – States). The whole list consists of more than 6 thousand pairs obtained automatically from the morphological synthesizer for Czech, manually checked and slightly extended.

### 4 Inter-Annotator Agreement

Several annotators annotated the data but (for obvious reasons of limited resources) each part of the data has only been annotated by one of them. Only 4% of the data (44 documents, 2,084 sentences) have been annotated in parallel by two annotators of discourse relations, and 3% (39 documents, 1,606 sentences) have been annotated in parallel by two annotators of textual coreference and bridging anaphora. We used the parallel (double) annotations for measuring the inter-annotator agreement, and for analyzing the most common errors, i.e. difficult parts of the annotation.

<sup>7</sup> This decision is considered to be provisional. The antecedents are supposed to be specified in further phases of the annotation.

<sup>6</sup> For more details, see Jínová et al. (2012b).

To evaluate the inter-annotator agreement on texts annotated in parallel by two annotators, we used several measures. The connective-based F1-measure (Mírovský et al., 2010c) was used for measuring the agreement on the recognition of a discourse relation, the chain-based F1-measure was used for measuring the agreement on the recognition of a coreference or bridging relation. A simple ratio and Cohen's  $\kappa$  were used for measuring the agreement on the type of the relations in cases where the annotators recognized the same relation.<sup>8</sup>

In the connective-based measure, we consider the annotators to be in agreement on recognizing a discourse relation if the two connectives they mark (each of the connectives marked by one of the annotators) have a non-empty intersection (technically, a connective is a set of tree nodes). For details, see Jínová et al. (2012a).

In the chain-based measure, we consider the annotators to be in agreement on recognizing a coreference or a bridging relation if two nodes connected by an arrow by one of the annotators have also been connected by the other annotator; coreference chains are taken into account, i.e. it is sufficient for the agreement if the arrow starts in or goes to a node that is coreferentially connected (possibly transitively) with the node used for the relation by the other annotator.

Table 2 shows the results of the inter-annotator agreement measurements.

relation	F1	agreement on types	Cohen's $\kappa$
discourse	0.83	0.77	0.71
text. coref.	0.72	0.90	0.73
bridging	0.46	0.92	0.89

Table 2: Inter-annotator agreement

Comparison of the inter-annotator agreement with other similar projects is difficult, as the projects usually use different annotation schemes and different scores. Nevertheless, some comparisons can be done:

The simple ratio agreement on types in discourse relations (0.77 on all parallel data, the third column of Table 2) is the closest measure to the way of measuring the inter-annotator agreement used on subsenses in the Penn Discourse Treebank 2.0, reported in Prasad et al. (2008). Their agreement was 0.8.

<sup>8</sup> In all our measurements, only inter-sentential discourse relations have been counted, as the intra-sentential relations were mostly annotated automatically.

In the annotation of coreference relations in OntoNotes, the inter-annotator agreement on English was 80.9 for newspaper texts and 78.4 for magazine texts. On Chinese, the agreement was 73.6 for newspaper texts and 74.9 for magazine texts (reported in Pradhan et al. 2012). These numbers can be compared with our chain-based F1 measure (0.72 in the second column of Table 2), as it is similar to the MUC-6 score they used.

As to the bridging anaphora, we can compare our chain-based F1 score (0.46 in the second column of Table 2) to F1 score on recognition of bridging relations reported for the annotation of the COREA corpus (Dutch texts); their agreement on newspaper texts was 0.39 (reported in Hendrickx et al., 2011).

## 5 The Corpus in Numbers<sup>9</sup>

Table 3 shows total numbers of annotated relations in the whole data of PDiT.

relation	count
discourse relations	20,542
- discourse inter-sentential	6,195
- discourse intra-sentential	14,347
textual coreference	87,299
grammatical coreference <sup>10</sup>	23,272
bridging anaphora	33,154

Table 3: Total numbers of annotated relations in PDiT

bridging type	count
ANAF	847
CONTRAST	2,305
FUNCT_P	516
PART_WHOLE	2,017
P_FUNCT	1,743
REST	2,226
SET_SUB	13,106
SUB_SET	5,885
WHOLE PART	4,509
<b>total</b>	<b>33,154</b>

Table 4: Distribution of bridging types in PDiT

In addition to the numbers in Table 3, there have been annotated 445 members of lists, 4,188 headings, 1,505 coreference relations to segment and 689 references out of the text (exophora).

<sup>9</sup> Please note that 1/10 of the PDT/PDiT data has been designated to evaluation tests. Numbers presented in this section include also this part of the data. Therefore, these numbers should not be used in any experiments tested on the evaluation test data of PDT/PDiT!

<sup>10</sup> mostly annotated already in PDT

Table 4 shows a distribution of bridging types annotated in PDiT. Table 5 shows the total number of individual discourse types annotated in PDiT.

discourse type	full name	count
conc	concession	878
cond	condition	1,369
confr	confrontation	654
conj	conjunction	7,551
conjalt	conj. alternative	90
corr	correction	440
disjalt	disj. alternative	270
equiv	equivalence	104
exempl	exemplification	142
explicit	explication	225
f_cond	pragm. condition	16
f_opp	pragm. contrast	50
f_reason	pragm. reason	40
gener	generalization	106
grad	gradation	430
opp	opposition	3,209
preced	asynchronous	808
purp	purpose	414
reason	reason-result	2,626
restr	restr. opposition	269
spec	specification	627
synchr	synchronous	222
<i>other</i>	<i>other</i>	2
<b>total</b>		<b>20,542</b>

Table 5: Distribution of discourse types in PDiT

## 6 Discussion

In the first release of PDiT, the annotation of discourse relations is limited to relations expressed by explicit DCs (coordinating conjunctions, particles, adverbs etc.), other tags between adjacent sentences were not inserted, unlike in some similar projects. Alternative lexicalizations (AltLex) are not annotated in PDiT, their thorough analysis is a recent work in progress. Entity-based relations (EntRel) are, in our view, a matter of coreference and bridging annotation.

### Implicit connectives

Annotation of implicit connectives has been in all known attempts a problematic task, as the IAA numbers are rather low. For implicit connectives (not present on the surface, a DC must be "inferred" from the context), we conducted an experimental annotation of 100 sentences, trying to remove factors known as repeatedly disturbing.<sup>11</sup> The annotators agreed in 49% on type of

<sup>11</sup> The annotation was carried out by two most experienced annotators, the chosen text types were from an accessible domain (cultural event description), the texts were short, up

to 35 sentences each. Another option would be to underspecify the sense hierarchy but we did not do that. Instead, we allowed for labels coref, bridging (=EntRel) and NoRel.

the relation. If only the distinction between *any* discourse relation on one side and coref + bridging relation on the other side was taken into consideration, the agreement was slightly higher – 58%. The most problematic issue revealed to be distinguishing between elaborative relations and relations based only on coreference. The restriction of the annotation only to slots between adjacent sentences was found useful for simplifying the annotation but it did not always match the annotators' intuition where the argument borders should be (e.g. if only the sentence-last dependent clause relates to the following sentence). Although the annotators were able to agree in most cases after discussion, the results convinced us to reconsider the annotation setting for implicit DCs before any future annotation.

Another phenomenon not present in PDiT in comparison with PDTB is attribution. We believe that this information can be at least partially obtained from syntactic features of the syntactic layers of PDT (e.g. attributes for direct speech, parentheses, verbal valency etc.).

## 7 Conclusion

We described the Prague Discourse Treebank 1.0, PDiT 1.0, a large collection of Czech texts that offers a rare combination of manual annotations of discourse relations, textual coreference and bridging anaphora. PDiT 1.0 is an extension of PDT 2.5 and all the annotation presented in this paper was carried out on the dependency trees of the tectogrammatical (deep syntax) layer. It was released in November 2012 under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License and it is available at the LINDAT-Clarin repository<sup>12</sup> (Poláková et al., 2012b).

Recently, we focus on extensions of the annotation for the upcoming release of PDT 3.0. A genre classification of the corpus texts for the purposes of data clustering in automatic experiments has been finished. Annotation of alternative lexicalizations (AltLex) and anaphoric expressions of 1st and 2nd person are in progress.

### Acknowledgment

We gratefully acknowledge support from the Grant Agency of the Czech Republic (projects n.

<sup>12</sup> <http://hdl.handle.net/11858/00-097C-0000-0008-E130-A>

P406/12/0658 and P406/2010/0875), the LIND-AT-Clarín project (LM2010013) and SVV of the Charles University (267 314).

## References

- S. D. Afantenos, N. Asher, F. Benamara et al. 2012. An empirical resource for discovering cognitive principles of discourse organization: the ANNODIS corpus. In: *Proceedings of LREC 2012*, Istanbul, Turkey.
- A. Al-Saif, K. Markert. 2010. The Leeds Arabic Discourse Treebank: Annotating discourse connectives for Arabic. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, Malta, pp. 2046–2053.
- E. Bejček, J. Panevová, J. Popelka et al. 2012. Prague Dependency Treebank 2.5 – a revisited version of PDT 2.0. In: *Proceedings of the 24th International Conference on Computational Linguistics (Coling 2012)*, Mumbai, India, pp. 231–246.
- L. Carlson, D. Marcu, M. E. Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue*, Eurospeech 2001.
- H. H. Clark. 1975. Bridging. In: *The Conference on Theoretical Issues in NLP*, pp. 169–174.
- L. Danlos, D. Antolinos-Basso, C. Braud et al. 2012. Vers le FDTB: French Discourse Tree Bank In: *Actes de la conférence conjointe JEP-TALN-RE-CITAL*, Grenoble, France, volume 2 : TALN, 2, pp. 471–478.
- G. Doddington, A. Mitchell, M. Przybocki et al. 2004. The Automatic Content Extraction (ACE) program – tasks, data, and evaluation. In: *Proceedings of LREC 2004*, Lisbon.
- A. Gastel, S. Schulze, Y. Versley et al. 2011. Annotation of Explicit and Implicit Discourse Relations in the TüBa-D/Z Treebank. In: *Multilingual Resources and Multilingual Applications, Proceedings of the German Society of Computational Linguistics and Language Technology (GSCL) 2011*. Hamburg, pp. 99–104.
- J. Hajič, J. Panevová, E. Hajičová et al. 2006. *Prague Dependency Treebank 2.0*. Software prototype, Linguistic Data Consortium, Philadelphia, PA, USA, ISBN 1-58563-370-4, <http://www ldc.upenn.edu>, Jul 2006.
- I. Hendrickx, O. De Clercq, V. Hoste. 2011. Analysis and Reference Resolution of Bridge Anaphora across Different Text Genres. In: *Anaphora Processing and Applications*. Lecture Notes in Computer Science Volume 7099, pp. 1–11.
- E. Hinrichs, S. Kübler, K. Naumann et al. 2004. Recent developments in linguistic annotations of the TüBa-D/Z treebank. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories*. Tübingen.
- L. Hirschman, N. Chinchor. 1997. *MUC-7 Coreference Task Definition – Version 3.0*.
- Y. Hou, K. Markert, M. Strube. 2013. Integrating semantics and saliences for bridging resolution using Markov logic. In *NAACL 2013* to appear.
- P. Jínová, J. Mirovský, L. Poláková. 2012a. Analyzing the Most Common Errors in the Discourse Annotation of the Prague Dependency Treebank. In: *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories (TLT 11)*, Lisbon, Portugal, November 2012.
- P. Jínová, J. Mirovský, L. Poláková. 2012b. Semi-Automatic Annotation of Intra-sentential Discourse Relations in PDT. In: *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), ADACA Discourse Workshop*, Mumbai, India, December 2012.
- I. Korzen, M. Buch-Kromann. 2011. Anaphoric relations in the Copenhagen dependency treebanks. In: *Beyond Semantics: Corpus-based Investigations of Pragmatic and Discourse Phenomena*. DGfS Workshop, pp. 83–98.
- O. Krasavina, Ch. Chiarcos. 2007. PoCoS –Potsdam Coreference Scheme. In *Proceedings of the Linguistic Annotation Workshop*, Prague.
- L. Kučová, E. Hajičová. 2004. Coreferential Relations in the Prague Dependency Treebank. In *Proceedings of the 5th Discourse Anaphora and Anaphor Resolution Colloquium*, S. Miguel.
- W.C. Mann, S. A. Thompson. 1988. Rhetorical structure theory. Toward a functional theory of text organization. In: *Text*, 8(3):243–281.
- M. Mikulová et al. 2005. *Annotation on the tectogrammatical layer in the Prague Dependency Treebank. The Annotation Guidelines*. Prague: UFAL MFF. Available at: <http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/index.html>.
- E. Miltsakaki, L. Robaldo, A. Lee et al. 2008. Sense Annotation in the Penn Discourse Treebank. In: *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*.
- J. Mirovský, L. Mladová, Z. Žabokrtský. 2010a. Annotation Tool for Discourse in PDT. In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Tsinghua University Press, Beijing, China, ISBN 978-7-302-23456-2, pp. 9–12.
- J. Mirovský, P. Pajas, A. Nedoluzhko. 2010b. Annotation Tool for Extended Textual Coreference and Bridging Anaphora. In: *Proceedings of the 7th International Conference on Language Resources*



- and Evaluation (LREC 2010), Valletta, Malta, ISBN 2-9517408-6-7, pp. 168–171.
- J. Mirovský, L. Mladová, Š. Zikánová. 2010c. Connective-Based Measuring of the Inter-Annotator Agreement in the Annotation of Discourse in PDT. In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Tsinghua University Press, Beijing, China, pp. 775–781.
- A. Nedoluzhko, J. Mirovský, M. Novák. 2013. A Coreferentially annotated Corpus and Anaphora Resolution for Czech. To appear in *Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue 2013"*. Moskva.
- A. Nedoluzhko, J. Mirovský. 2011. *Annotating Extended Textual Coreference and Bridging Relations in the Prague Dependency Treebank*. Technical report no. 2011/44, ÚFAL MFF UK, Prague, Czech Republic, 69 pp.
- U. Oza, R. Prasad, S. Kolachina et al. 2009. The Hindi Discourse Relation Bank. In: *Proc. Linguistic Annotation Workshop*, pp.158–161.
- P. Pajas, J. Štěpánek. 2008. Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, pp. 673–680.
- M. Poesio, R. Vieira, S. Teufel. 1997. Resolving bridging references in unrestricted text. In: *ACL Workshop on Robust Anaphora Resolution*, pp. 1–6.
- M. Poesio. 2004. The MATE/GNOME Proposals for Anaphoric Annotation, Revisited. In: *Proceedings of The 5th SIGdial Workshop on Discourse and Dialogue*, Boston.
- M. Poesio, R. Delmonte, A. Bristot et al. 2004a. *The Venex corpus of anaphora and deixis in spoken and written Italian*. Manuscript.
- M. Poesio, R. Mehta, A. Maroudas et al. 2004b. Learning to resolve bridging references. In: *42nd Meeting of the Association for Computational Linguistics (ACL 2004)*, pp. 143–150.
- M. Poesio, R. Artstein. 2008. Anaphoric annotation in the ARRAU corpus. In *Proceedings of LREC 2008*, Marrakech.
- L. Poláková, P. Jínová, Š. Zikánová et al. 2012a. *Manual for Annotation of Discourse Relations in the Prague Dependency Treebank*. Technical report, ÚFAL MFF UK, Prague, Czech Republic. Available at: <http://ufal.mff.cuni.cz/techrep/tr47.pdf>.
- L. Poláková, P. Jínová, Š. Zikánová et al. 2012b. *Prague Discourse Treebank 1.0*. Data/software, ÚFAL MFF UK, Prague, Czech Republic, <http://ufal.mff.cuni.cz/discourse/>, Nov 2012.
- S. Pradhan, E. Hovy, M. Marcus et al. 2007. Ontonotes: A unified relational semantic representation. In: *Proceedings of the International Conference on Semantic Computing*, Washington DC.
- S. Pradhan, A. Moschitti, N. Xue et al. 2012. *CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes*. Jeju, South Korea, Jul 2012.
- R. Prasad, N. Dinesh, A. Lee et al. 2008. The Penn Discourse Treebank 2.0. In: *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, pp. 2961–2968.
- R. Prasad, S. McRoy, Nadya Frid et al. 2011. *The Biomedical Discourse Relation Bank, BMC 1*, 12:188
- M. Recasens, A. M. Martí. 2009. AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. In: *Language Resources and Evaluation*.
- K. Rodríguez, F. Delogu, Y. Versley et al. 2010. Anaphoric Annotation of Wikipedia and Blogs in the Live Memories Corpus. In *Proceedings of LREC 2010*, Valletta, Malta.
- M. Stede. 2004. The Potsdam Commentary Corpus. *Proc. of the ACL 2004 Workshop on Discourse Annotation*, pp. 96–102.
- S. Tonelli, G. Riccardi, R. Prasad et al. 2010. Annotation of Discourse Relations for Conversational Spoken Dialogs. 2010. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pp. 2084–2090. Valletta, Malta.
- B. Webber, A. Knott, M. Stone et al. 2003. Anaphora and Discourse Structure. *Computational Linguistics 29(4)*, pp. 545–588.
- F. Wolf, E. Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2).
- D. Zeyrek, I. Demirşahin, A. Sevdik-Çalli et al. 2010. The Annotation Scheme of the Turkish Discourse Bank and an Evaluation of Inconsistent Annotations. In: *Proceedings of the Fourth Linguistic Annotation Workshop*. Pages 282–289. Uppsala, Sweden.
- Y. Zhou, N. Xue. 2012. PDTB-style Discourse Annotation of Chinese Text. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. pp. 69–77. Jeju, Republic of Korea. July 2012.



# Multilingual Mention Detection for Coreference Resolution

**Olga Uryupina**

DISI, University of Trento, Italy  
uryupina@gmail.com

**Alessandro Moschitti**

QCRI, Qatar Foundation, Doha, Qatar  
amoschitti@qf.org.qa

## Abstract

This paper proposes a novel algorithm for multilingual mention detection: we extract mentions from parse trees via kernel-based SVM learning. Our approach allows for straightforward mention detection for any language where (not necessary perfect) parsing resources are available, without any complex language-specific rule engineering. We also investigate possibilities for incorporating automatically acquired mentions into an end-to-end coreference resolution system. We evaluate our approach on the Arabic and Chinese portions of the CoNLL-2012 dataset, showing a significant improvement over the system with the baseline mention detection.

## 1 Introduction

Accurate mention detection (MD) is a vital prerequisite for a variety of Natural Language Processing tasks, in particular, for Relation Extraction (RE) and Coreference Resolution (CR). If a toolkit cannot extract mentions reliably, it will obviously be unable to assign them to relations or entities.

Many studies on RE and CR report evaluation figures on *gold* mentions: in such a setting, a system is supplied with correct mention boundaries and/or semantic classes or other relevant properties. It can, in theory, be argued that such a methodology provides better insights on performance of RE and CR algorithms per se. It has been demonstrated, however, that evaluation results on gold mentions are misleading: for example, Ng (2008) shows that unsupervised CR algorithms exhibit promising results on gold mentions, that are not mirrored in a more realistic evaluation on automatically detected mentions.

The exact scope of the mention detection task varies considerably depending on the annotation

guidelines. Thus, some corpora consider all the (non-embedding) NPs to be mentions, some corpora do not allow for non-referential mentions and some do not mark singleton referential mentions, that do not participate in coreference relations. In addition, some guidelines may restrict the annotation to specific semantic types.

A number of linguistic studies focus on various syntactic, semantic and discourse clues that might help identify nominal constructions that cannot participate in coreference relations. Possible features include, among others, specific syntactic constructions for expletive pronouns, negation, modality and quantification (Karttunen, 1976). Several algorithms have been proposed recently, trying to tackle some of the addressed phenomena within a computational approach. Thus, a number of algorithms have been developed recently to identify expletive usages of “it” (Evans, 2001; Boyd et al., 2005; Bergsma and Yarowsky, 2011). While these approaches are potentially beneficial for mention detection in English, for other languages, neither theoretical nor computational studies are available at the moment. In this paper, we use tree kernels to extract relevant syntactic patterns automatically, without assuming any prior knowledge of the input language.

In this paper, we propose a learning-based solution to the mention detection task. We use SVMs (Joachims, 1999) with syntactic tree kernels (Collins and Duffy, 2001; Moschitti, 2008; Moschitti, 2006) to classify parse tree nodes as  $\pm$ mentions. Our approach does not require any language- or corpus-specific engineering and thus can be easily adapted to cover new languages or mention annotation schemes.

The rest of paper is organized as follows. In the next section, we define the task and discuss our tree and vector representations. Section 4 presents MD evaluation figures. Finally, in Section 5 we incorporate our MD module into an end-to-end

coreference resolution system.

## 2 Related Work

Until recently, most RE and CR toolkits have been evaluated on the ACE datasets (Dodgington et al., 2004). The ACE guidelines restrict possible mentions to be considered to specific semantic types (PERSON, LOCATION and so on). Moreover, mentions are annotated with their minimal and maximal span, allowing for relaxed matching between gold and automatically extracted boundaries. In such a setting, the mention detection task can be cast as a tagging problem, similar to the named entity recognition and classification task. A number of systems have followed this scenario, demonstrating reliable performance (Florian et al., 2004; Ittycheriah et al., 2003; Zitouni and Florian, 2008).

In the past years, however, several corpora have been created from a more linguistic perspective: for example, the OntoNotes dataset (Hovy et al., 2006; Pradhan et al., 2012) provides annotation for unrestricted coreference. The guidelines differ significantly from the ACE scheme: mentions correspond to parse nodes and can be of any semantic type, the systems are expected to recover mention boundaries exactly. The OntoNotes mentions—unlike ACE ones—correspond to large NP structures (embedding NP nodes in gold parse trees), so a traditional approach (e.g., one of those mentioned above), which aims at identifying basic NP chunks, would not be applicable here. Therefore, any MD method for OntoNotes would rely on parsing.

The OntoNotes corpus has been used for evaluating end-to-end CR systems at two CoNLL shared tasks (2011 and 2012). At the 2011 shared task, the participants relied on rule-based modules for extracting mention boundaries from parse trees. This was relatively straightforward, as the task was devoted to CR in English and most participants could use their in-house MD modules developed and refined in the past decade. At the 2012 shared task, however, the systems were expected to provide end-to-end coreference resolution for Arabic and Chinese. As it turned out, most groups could not adapt their MD rules to cover these two languages and fell back to very simple baselines (e.g., “use all NP nodes as mentions”). Kummerfeld et al. (2011) investigated various post- and pre-filtering heuristics for

adapting their mention detection algorithm to the OntoNotes English data in a semi-automatic way, reporting mixed results.

## 3 Mention extraction from parse trees

We recast MD as a *node filtering* task: each candidate node is classified as either mention or not. In this study, we consider all “NP” nodes to be candidates for MD. As Table 1 shows, this is a reasonable assumption for the OntoNotes dataset, as almost 90% of all the mentions for both Arabic and Chinese correspond to NP nodes. The remaining 11-14% of mentions can mostly be attributed to parsing errors: as we aim at end-to-end processing with no gold information available, we run our system on automatically extracted parse trees, it is therefore possible that a mention corresponds to a gold NP node that has not been labeled correctly in an automatic parse tree.

	train		development	
	NP-nodes	%	NP-nodes	%
ARB	24068	87.23	2916	87.91
CHN	88523	85.96	12572	88.52

Table 1: NP-nodes in OntoNotes for Arabic (ARB) and Chinese (CHN): total numbers and percentage of mentions that are NP-nodes.

Not all the NP nodes, however, correspond to a mention. Such non-mention NPs fall into several categories:

- **Embedded NPs.** When an NP is embedded into another one, only the outer NP is used to represent a mention:

$$(1) \quad [MENTION-NP[NP \text{This type}] \text{ of earthquake}] \text{ has no precursors. }^1$$

A number of heuristics have been proposed for English to identify and discard embedded NPs, based on available head-finding algorithms, e.g., (Collins, 1999). For other languages, however, the task of finding a head of a given NP in a constituency tree is not trivial.

- **Non-referential NPs.** Depending on the annotation guidelines, non-referential NPs can

<sup>1</sup>We use English OntoNotes examples throughout this paper to illustrate discussed phenomena, as our approach is language-independent. The evaluation, however, is done on Arabic and Chinese.

either be marked as mentions or not. In OntoNotes, non-referential NPs should not be annotated:

(2) This type of earthquake has [ $NP$ no precursors].

- **Singleton NPs.** In some CR corpora (for example, ACE), mentions are annotated even if they do not participate in any coreference relations. In other corpora (MUC and OntoNotes), such *singletons* are not marked. When singletons are not marked, the MD tasks becomes considerably more difficult: the performance of an MD component cannot be measured and optimized directly, but only in conjunction with a coreference resolver.
- **Erroneous NPs.** When we evaluate an end-to-end system, we expect it to process raw input and thus rely on automatically extracted parse trees. Some NP-nodes might be incorrect, not corresponding to any NP in the gold tree. Such nodes cannot be mentions:

(3) At the meeting, Huang Xiangning read [ $NP$ the earthquake prediction] that they had previously issued.

“The earthquake prediction” is considered to be an NP node by the parser. In the gold data, however, this node does not exist at all. And even if it existed, the mention should correspond to its embedding NP node, “the earthquake prediction that the had previously issued” (cf. example 1 above). While this problem is less crucial for English, parsing resources for other languages are still scarce and less reliable.

### 3.1 Tree Representation

We use kernel-based SVMs to classify nodes as  $\pm$ mentions. This requires representing a relevant fragment of a tree with a specific node marked as “C-NP” (candidate). We start from a straightforward representation: using automatically generated parse trees provided within the CoNLL data distribution, we generate one example for each NP node: the example corresponds to the entire parse tree with just a single node re-labeled as “C-NP”. The assigned class label reflects the fact that this particular node corresponds to some gold mention or not. For example, the full parse tree for our sentence (1-2) will generate one positive (for “This

type of earthquake”, shown on Figure 1) and three negative examples (for “This type”, “earthquake” and “no precursors”).

While this representation might work for our toy example, for a longer sentence it would provide irrelevant information. Consider again the tree on Figure 1. To generate a training example we append “C-” to one NP node, keeping all the remaining nodes as-is. The tree kernel operates on subtrees of the given structure, so, effectively, it will consider a lot of tree fragments that do not contain the marked node. These fragments will affect the treatment of different examples, possibly with conflicting class labels. It will not only make learning slow but also introduce spurious evidence, decreasing the system’s performance. We have therefore investigated two possibilities for pruning our trees.

Our first pruning algorithm (“up-down”) starts from the node of interest (C-NP) and goes up for  $u$  nodes. From each node on the path, it considers all its children up to the depth  $d$ . The first part of Figure 2 shows a pruned tree for  $u = 2, d = 1$  for the node “This type of earthquake.”

Our second pruning algorithm (“radial”) starts from the node of interest and considers all the nodes in the tree that are reachable from it via at most  $n$  edges. The second part of Figure 2 shows a pruned tree for  $n = 2$  for the same node.

### 3.2 Vector Representation

In addition to (pruned) trees, we also provide vector representations of our NPs. For each NP, we extract its basic properties: number, gender, person, mention type (name, nominal or pronoun) and the number of other NPs in the document that have the same surface form. To extract mention properties, we have to compute the head. However, the goal of our study is to provide an MD algorithm that is adaptable to different languages without extensive engineering. We have therefore deliberately relied on an over-simplistic heuristic for finding an NP head: either the last or the first noun in an NP is considered a head, depending on some very basic information on a word order in a specific language. Given the head, we extract its properties from the CoNLL data in a straightforward way (for example, we have compiled a list of pronouns with their gender, number and person values from the training data and so on). This is done fully automatically and doesn’t require any

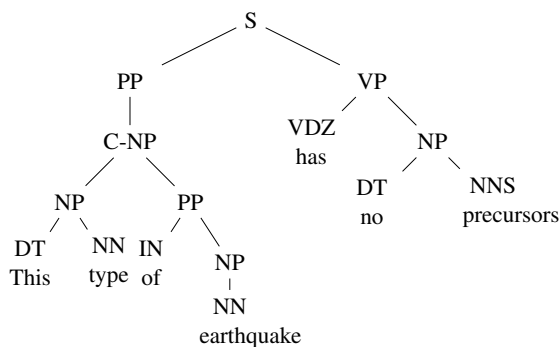


Figure 1: Parse tree for “This type of earthquake”, examples (1-2): before pruning.

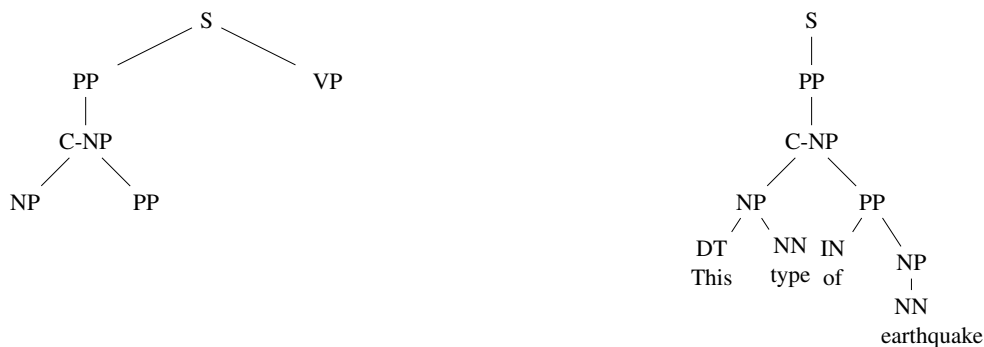


Figure 2: Up-down (left) vs. radial (right) pruning for “This type of earthquake,” examples (1-2)

language-specific manual engineering.

Table 2 lists the features for our vector representation. Nominal values are binarized, leading to 10 binary or continuous features.

feature	possible values
Gender	F,M,Unknown
Definiteness	Yes, No
Number	Sg,Pl,Du,Unknown
MentionType	Name,Nominal,Pronoun
#same-surface NPs in the doc	continuous (normalized)

Table 2: Features used for Mention Detection: each feature describes an individual NP

## 4 Evaluating MD

In this section we provide evaluation results on both Arabic and Chinese. We reserve a small portion of the CoNLL training data (around 20k instances for each language) for training an MD system. Another small subset (around 5k instances) is reserved for fitting the system parameters. The evaluation results are reported on the CoNLL development data. Note that we evaluate the NP-

node classifier, so the system receives no penalty for missing mentions that are not NPs. In Section 5 below, however, we will assess the impact of MD on the end-to-end CR system and thus penalize for missing non-NP mentions.

As a baseline (“all-NP”), we consider all the NP nodes to be mentions. Table 3 below compares this baseline against mentions extracted automatically from different representations. We use Syntactic Tree Kernels (TK) implemented within the SVM-TK toolkit<sup>2</sup> to induce the classification.

As our results suggest, vector representation does not provide enough information for robust mention detection.<sup>3</sup> Indeed, without tree kernels, the system is only able to learn a major class labeling. This highlights the importance of a model that is able to handle structured input, learning relevant patterns directly from parse trees.

As discussed in Section 3 above, full trees contain too much misleading evidence. A single parse tree might contain several dozens of NP nodes, so,

<sup>2</sup><http://disi.unitn.it/moschitti/Tree-Kernel.htm>

<sup>3</sup>As a pilot experiment, we also added bag-of-words features to our vector representations, but this didn’t yield any improvement.

representation	pruning	R	P	F
Arabic				
all-NP	N/A	100	18.0	30.5
vectors	N/A	0	N/A	N/A
trees	-	53.9	45.5	49.4
trees	d=3, u=1	59.5	50.4	54.6
trees	n=2	59.6	64.7	62.0
trees+vectors	d=3, u=1	65.1	52.9	58.4
trees+vectors	n=2	66.0	66.1	66.1
Chinese				
all-NP	N/A	100	27.9	43.6
vectors	N/A	0	N/A	N/A
trees	-	65.0	55.8	60.0
trees	d=1, u=1	73.0	56.9	63.9
trees	n=2	69.7	63.7	66.6
trees+vectors	d=1, u=1	75.6	60.7	67.4
trees+vectors	n=3	71.3	68.9	70.1

Table 3: Performance of the MD classifier on the development set

using a full sentence tree to represent a particular candidate node provides confusing input for the classifier. This is reflected with a low classifier performance on full representations.

Both pruning strategies have resulted in a substantial improvement in the performance level. The radial pruning has significantly outperformed the up-down strategy. Moreover, the radial pruning depends on just one parameter and can therefore be optimized faster.

Finally, joint vector and tree representation further outperforms a plain tree-based model. It must be noted, however, that our MD features (Table 2) require at least some minimal amount of language-specific engineering.

## 5 Incorporating TK-based Mention Detection into an end-to-end coreference resolution system

For our experiments, we use BART – a modular toolkit for coreference resolution that supports state-of-the-art statistical approaches to the task and enables efficient feature engineering (Versley et al., 2008). BART has originally been created and tested for English, but its flexible modular architecture ensures its portability to other languages and domains.

In our evaluation experiments, we follow a very simple model of coreference, namely, the mention-pair approach advocated by Soon et al.

(2001) and adopted in many studies ever since. We believe, however, that more complex models of coreference will also benefit from our MD algorithm: most state-of-the-art CR systems treat mention detection as a preprocessing step that is not affected by further processing and therefore we expect them to yield better performance when such a preprocessing is achieved in a more robust way.

Creating a robust coreference resolver for a new language requires linguistic expertise and language-specific engineering. This cannot and, moreover, should not be avoided by fully language-agnostic methods. Our approach to end-to-end coreference resolution relies on a universal MD component that requires no linguistic engineering – it facilitates the development of coreference resolvers in the narrow sense, by providing them with input mentions. We must stress that the resolvers themselves are not supposed to be universal: in fact, a number of linguistic studies on coreference address various language-specific challenging problems (e.g., zero pronouns, different marking of information status etc).

Below we describe the adjustments we made to BART to cover Arabic and Chinese and then report on our experiments for integrating kernel-based MD into BART to provide an end-to-end coreference resolution for these languages.

### 5.1 Adapting BART to Arabic and Chinese

The modularity of the BART toolkit enables its straightforward adaptation to different languages. This includes creating meaningful linguistic representations of mentions (“mention properties”) and, optionally, some experiments on feature selection and engineering.

We extracted some properties (sentence boundaries, lemmata, speaker id) for Arabic and Chinese directly from the CoNLL/OntoNotes layers<sup>4</sup>. Mention types are inferred from PoS tags.

We compiled lists of pronouns for both Arabic and Chinese from the training and development data. For Arabic, we used gold PoS tags to classify pronouns into subtypes, person, number and gender. For Chinese, no such information is available, so we consulted several grammar sketches and lists of pronouns on the web. Finally, we extracted a list of gender affixes for Arabic along

<sup>4</sup>Recall that all the layers, apart from the Arabic lemma, were computed using state-of-the-art preprocessing tools by the CoNLL organizers and do not contain gold information

with a list of gender-classified lemmata from the training data.

We assessed the list of features, supported by BART, discarding those that require unavailable information (for example, the `aliasing` feature relies on semantic types for named entities that are not available within the CoNLL/OntoNotes distribution for languages other than English). We also created two additional features: `LemmataMatch` (similar to string match, but uses lemmata instead of tokens) and `NumberAgreementDual` (similar to commonly used number agreement features, but supports dual number). Both features are expected to provide important information for coreference in Arabic, a morphologically rich language.

We ran a feature selection experiment to further remove irrelevant features (BART were only tested on European languages, thus several features reflected patterns more common for Germanic and Romance languages). This resulted in two feature sets, one for each language, listed in Table 4. For comparison, we also show the baseline features (cf. below).

## 5.2 Incorporating Kernel-based MD into a Coreference Resolver

Coreference resolution systems have different tolerance for precision and recall MD errors. If a spurious mention is introduced, the CR system might still assign it to no coreference chain and thus discard from the output partition. If a correct mention is missed, however, the system has no chance of recovering it as it does not even start processing such a mention. This suggests that an MD module should be tuned to yield better recall.

To assess the impact of MD precision and recall errors on the performance of our coreference resolver, we run a simulation experiment. We start from the upper bound baseline: the MD module considers all the true (gold) NP mentions to be positive and all the spurious ones – to be negative. We then randomly distort this baseline, adding spurious mentions and removing correct ones, to arrive at a predefined performance level. The resulting MD output is then sent to our coreference resolution system and its performance is measured. As a measure of the CR system performance, we use the MELA F-score – an average of MUC,  $B^3$  and  $CEAF_e$  metrics, the official performance measure at the CoNLL shared task (Pradhan et al., 2012).

Figure 3 shows the results of our simulation experiment on the development data. Each line on the figure corresponds to a single MD recall level (varying from 100% to 70%). On the horizontal axis, we plot the MD precision (from 10% to 100%) and on the vertical axis – the end-to-end system MELA F-score. The curves support our intuition that reliable MD recall is crucial for coreference: when the MD recall drops to around 70%, the MELA score remains at the baseline level even for very high MD precision. It must be noted that our simulation experiment relies on an unrealistic assumption: we assume all the errors to be independent. In a more practical setting, the MELA F-score for a given combination of MD precision and recall can be higher, because the coreference system might fail to resolve the same NPs that are problematic for the MD module. Nevertheless, the curves illustrate the fact that any MD module should be strongly biased towards recall in order to be useful for coreference resolution.

We therefore reran our optimization experiments to fit more parameters of the MD module. Recall from Section 4 that we already used a small amount of CoNLL training data to fit our  $d$ ,  $u$  and  $n$  values. We expanded the set of parameters, using the end-to-end performance (MELA F-score) to select optimal values on the same subset. Table 5 lists all the parameters of our MD module.

d, u	up-down pruning thresholds
n	radial pruning threshold
j	precision-recall trade-off (SVM-TK)
c	cost factor (SVM-TK)
s	size of MD vs. CR data splits
r	tree vs. tree+vector representation

Table 5: Parameters optimized on a held-out data

Our experiments reveal that, indeed, a recall-oriented version of our MD classifier yields the most reliable end-to-end resolution. Table 6 shows the MD performance of the best classifier selected according to the MELA score. While the F-scores of these biased classifiers are, obviously, much lower than their unbiased counterparts, they still manage to filter out a substantial amount of noun phrases, at the same time maintaining a very high recall level.

Finally, Tables 7 and 8 show the MELA score of BART on the CoNLL-2012 development and test sets respectively. To evaluate the impact of our

feature		Baseline	Arabic	Chinese
StringMatch	$M_i$ and $M_j$ have the same surface form	+	+	+
MentionType	relevant types of $M_i$ and $M_j$ , (cf. Soon et al.)	+	+	+
GenderAgree	$M_i$ and $M_j$ agree in gender	+	+	+
NumberAgree	$M_i$ and $M_j$ agree in number	+		+
NumberAgreeDual	-*- , supports dual number		+	
AnimacyAgree	$M_i$ and $M_j$ agree in animacy	+		+
Compatible	$M_i$ and $M_j$ don't disagree or overlap		+	
Alias	heuristic NE-matching	+		
DistanceSentence	distance in sentences between $M_i$ and $M_j$	+		+
Appositive	$M_i$ and $M_j$ are in an apposition	+		
First_Mention	$M_i$ is the first mention in its sentence		+	
DistanceMarkable	distance in mentions between $M_i$ and $M_j$		+	
FirstSecondPerson	$M_{i/j}$ is a pronoun of the 1st/second person		+	
NonProSalience	for non-pro $M_i$ , # preceding same-head mentions		+	
SpeakerAlias	heuristics for 1/2 pers. pro, use "speaker" layer			+

Table 4: Features used for Coreference Resolution in Arabic and Chinese: each feature describes a pair of mentions  $\{M_i, M_j\}$ ,  $i < j$ , where  $M_i$  is a candidate antecedent and  $M_j$  is a candidate anaphor

kernel-based MD (TKMD), we compare its performance against two baselines. The lower bound, "all-NP", considers all the NP-nodes in a parse tree to be candidate mentions. The upper bound, "gold-NP" only considers gold NP-nodes to be candidate mentions. Note that the upper bound does not include mentions that do not correspond to NP-nodes at all (around 12% of all the mentions in the development data, cf. Table 1 above).

Tables 7 and 8 also show the performance level of BART's rule-based MD module that was developed for English. Although this heuristic has proved reliable on the English data, for example, at the CoNLL 2011 and 2012 shared tasks, it is not robust enough to be ported as-is to other languages: indeed, the performance of the heuristic MD on Arabic and Chinese is lower than the all-NP baseline. This highlights the importance of a learning-based approach: while rule-based MD shows good results for English, we cannot expect spending ten more years on designing similar systems for other languages.

	R	P	F
Arabic	90.67	31.07	46.28
Chinese	98.37	38.27	55.1

Table 6: Performance of the recall-oriented MD classifier on the CoNLL development set.

For both languages, the performance goes up

	Soon et al. (2001) features	Table 4 features
Arabic		
all-NP	46.15	46.32
English MD	43.46	43.49
TK-MD	48.13 <sup>†</sup>	50.02 <sup>†</sup>
gold-NP	63.27 <sup>†</sup>	64.55 <sup>†</sup>
Chinese		
all-NP	51.04	51.40
English MD	46.77	46.77
TK-MD	53.40 <sup>†</sup>	53.86 <sup>†</sup>
gold-NP	57.30 <sup>†</sup>	57.98 <sup>†</sup>

Table 7: Evaluating the impact of MD and linguistic knowledge: MELA F-score on the development set, significant improvement over the corresponding all-NP baseline shown with <sup>†</sup>.

drastically when one shifts from a realistic evaluation (the "all-NP" baseline) to gold NP mentions. Kernel-based MD is able to recover part of this difference, providing significant improvements over the baseline (t-test on individual documents,  $p < 0.05$ ).

Another important point is the difference between our basic feature set and more specific features (cf. Table 4). The contribution of extra features is relatively small and not significant, which is not surprising given the fact that all of them are very naive and do not address any coreference-

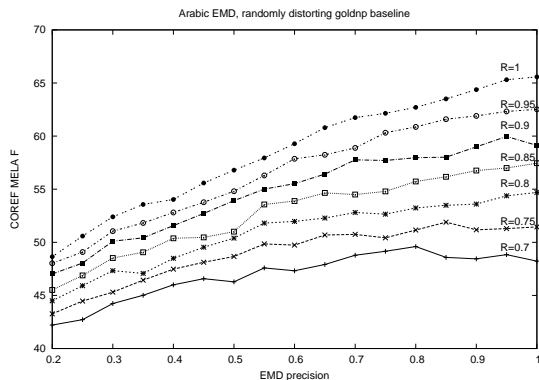


Figure 3: Performance of an end-to-end coreference resolution system for different values of MD Recall and Precision in a simulation experiment: MELA F-score on the Arabic and Chinese development data.

related phenomena specific for Arabic and Chinese. However, the extra features help more when the MD improves. This suggests that a robust MD module is an essential prerequisite for further work on coreference in new languages: a more accurate set of mentions provides a better testbed for manually engineered language-specific features or constraints.

## 6 Conclusion and Future Work

In this paper we have investigated possibilities for language-independent mention detection based on syntactic tree kernels. We have shown that a kernel-based approach can provide a robust pre-processing system that is a vital prerequisite for fast and efficient development of end-to-end multilingual coreference resolvers.

We have evaluated different tree and vector representations, showing that the best performance is

	Soon et al. (2001) features	Table 4 features
Arabic		
all-NP	46.79	47.36
English MD	43.77	43.65
TK-MD	48.38 <sup>†</sup>	51.54 <sup>†</sup>
gold-NP	63.07 <sup>†</sup>	65.57 <sup>†</sup>
Chinese		
all-NP	53.26	53.24
English MD	48.99	48.99
TK-MD	58.11 <sup>†</sup>	58.15 <sup>†</sup>
gold-NP	59.97 <sup>†</sup>	60.04 <sup>†</sup>

Table 8: Evaluating the impact of MD and linguistic knowledge: MELA F-score on the official CoNLL-2012 test set, significant improvement over the corresponding all-NP baseline shown with <sup>†</sup>.

achieved by applying radial pruning to parse trees and augmenting the resulting representation with feature vectors, encoding very basic and shallow properties of candidate NPs.

We have investigated possibilities of incorporating our MD module to an end-to-end coreference resolution system. Our evaluation results show significant improvement over the system relying on the “all-NP” baseline for both Arabic and Chinese. It should be stressed that no other baseline is available without using deep linguistic expertise.

In the future, we plan to follow two directions to further improve our algorithm. First, we want to consider more global models of MD, providing joint inference over sets of NP nodes, and, possibly, incorporating CR predictions as well. Several studies (Daume III and Marcu, 2005; Denis and Baldridge, 2009) followed this direction recently, showing promising results for joint MD and CR modeling.

Second, we want to combine our learning-based MD with more traditional heuristic systems. While our approach provides a fast reliable testbed and allows CR researchers to specifically focus on coreference, rule-based MD modules have been created for a variety of languages, especially for European ones, in the past decade. We believe that by combining such systems with our kernel-based algorithm, we can build MD modules that show a high performance level and, at the same time, are more robust and portable to different domains and corpora.



## Acknowledgments

The research described in this paper has been partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under the grant #288024: LIMOSINE – Linguistically Motivated Semantic aggregation engines.

## References

- Shane Bergsma and David Yarowsky. 2011. NADA: A robust system for non-referential pronoun detection. In *Proc. DAARC*, pages 12–23, Faro, Portugal, October.
- Adriane Boyd, Whitney Gegg-Harrison, and Donna Byron. 2005. Identifying nonreferential *it*: A machine learning approach incorporating linguistically motivated patterns. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, pages 40–47.
- Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Pascal Denis and Jason Baldridge. 2009. Global joint models for coreference resolution and named entity classification. In *Procesamiento del Lenguaje Natural 42, Barcelona: SEPLN*.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program—tasks, data, and evaluation. In *Proceedings of the Language Resources and Evaluation Conference*.
- Richard Evans. 2001. Applying machine learning toward an automatic classification of *it*. *Literary and Linguistic Computing*, 16(1):45–57.
- Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Xiaoqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–8.
- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of HLT/NAACL*.
- Hal Daume III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing*.
- Abraham Ittycheriah, Lucian Vlad Lita, Nanda Kambhatla, Nicolas Nicolov, Salim Roukos, and Margo Stys. 2003. Identifying and tracking entity mentions in a maximum entropy framework. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.
- Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.
- Lauri Karttunen. 1976. Discourse referents. In J. McKawley, editor, *Syntax and Semantics*, volume 7, pages 361–385. Academic Press.
- Jonathan K Kummerfeld, Mohit Bansal, David Burkett, and Dan Klein. 2011. Mention detection: Heuristics for the OntoNotes annotations. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 102–106, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of European Conference on Machine Learning*, pages 318–329.
- Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceeding of the International Conference on Information and Knowledge Management*, NY, USA.
- Vincent Ng. 2008. Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 640–649.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistic*, 27(4):521–544.
- Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. BART: a modular toolkit for coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 9–12.
- Imed Zitouni and Radu Florian. 2008. Mention detection crossing the language barrier. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

# A Weakly Supervised Bayesian Model for Violence Detection in Social Media

**Elizabeth Cano Yulan He**

School of Engineering and Applied Science  
Aston University, UK

{a.cano\_basave, y.he9}@aston.ac.uk

**Kang Liu Jun Zhao**

Institute of Automation

Chinese Academy of Sciences, China

{kliu, jzhao}@nlpr.ia.ac.cn

## Abstract

Social streams have proven to be the most up-to-date and inclusive information on current events. In this paper we propose a novel probabilistic modelling framework, called violence detection model (VDM), which enables the identification of text containing violent content and extraction of violence-related topics over social media data. The proposed VDM model does not require any labeled corpora for training, instead, it only needs the incorporation of word prior knowledge which captures whether a word indicates violence or not. We propose a novel approach of deriving word prior knowledge using the relative entropy measurement of words based on the intuition that low entropy words are indicative of semantically coherent topics and therefore more informative, while high entropy words indicates words whose usage is more topical diverse and therefore less informative. Our proposed VDM model has been evaluated on the TREC Microblog 2011 dataset to identify topics related to violence. Experimental results show that deriving word priors using our proposed relative entropy method is more effective than the widely-used information gain method. Moreover, VDM gives higher violence classification results and produces more coherent violence-related topics compared to a few competitive baselines.

## 1 Introduction

Social media and in particular Twitter has proven to be a faster channel of communication when compared to traditional news media, as we have witnessed during events such as the Middle East revolutions and the 2011 Japan earthquake; acting as social sensors of real-time events (Sakaki et al., 2010). Therefore the identification of topics discussed in these channels

could aid in different scenarios including violence detection and emergency response. In particular the task of classifying tweets as violence-related poses different challenges including: high topical diversity; irregular and ill-formed words; event-dependent vocabulary characterising violence-related content; and an evolving jargon emerging from violent events.

Indeed, machine learning methods for classification present difficulty on short texts (Phan et al., 2008). A large body of work has been proposed for the task of topic classification of Tweets (Milne and Witten., 2008; Gabrilovich and Markovitch, 2006; Genc et al., 2011; Muñoz García et al., 2011; Kasiviswanathan et al., 2011; Meij et al., 2012). Recent approaches have also been proposed (Michelson and Macskassy, 2010; Cano et al., 2013), to alleviate microposts sparsity by leveraging existing social knowledge sources (e.g Wikipedia). However, while the majority of these approaches rely on supervised classification techniques, others do not cater for the violence detection challenges. To the best of our knowledge very few have been devoted to violent content analysis of Twitter, and none has carried out deep violence-related topic analysis. Since violence-related events tend to occur during short to medium life spans, traditional classification methods which rely on labelled data can rapidly become outdated. Therefore in order to maintain tuned models it is necessary the continuous learning from social media in order to capture those features representing violent events. Indeed, the task of violence classification demands more efficient and flexible algorithms that can cope with rapidly evolving features. These observations have thus motivated us to apply unsupervised or weakly supervised approaches for domain-independent violence classification.

Another shortcoming of previous classification approaches is that they only focus on detecting the overall topical category of a document. However they do not perform an in-depth analysis to discover the latent topics and the associated document category.

When examining violence-related data, analysts are not only interested in the overall violence of one particular tweet but on the understanding of the type of emerging violence-related events. For example the word “killing” may have a violent-related orientation as in “mass killing” while it has a non-violent one in “killing time”. Therefore, detecting topic and violence-relatedness simultaneously should serve as a critical function in helping analysts by providing more informative violence-related topic mining results.

In this paper, we introduce the Violence Detection Model (VDM), which focuses on document-level violence classification for general domains in conjunction with topic detection and violence-related topic analysis. The model extends the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) by adding a document category (violent or non-violent) layer between the document and the topic layer. It is related to the joint sentiment-topic (JST) model for simultaneous sentiment and topic detection (Lin and He, 2009; Lin et al., 2012). However, while JST assumes the per-document sentiment-topic distributions, VDM only has a single document category-topic distribution shared across all the documents. This is because tweets are short compared to typical review documents and hence modelling per-tweet category-topic distribution could potentially generate less coherent topics. In VDM, we also assume that words are generated either from a category-specific topic distribution or from a general background model. This helps reducing the effects of background words and learn a model which better captures words concentrating around category-specific topics. As will be discussed later, VDM outperforms JST in both violence detection from tweets and topic coherence measurement. Furthermore, while JST incorporates word prior sentiment knowledge from existing sentiment lexicons, we propose a novel approach to derive word prior knowledge based on the relative entropy measurement of words.

We proceed with related work on topic classification on Twitter. Since the Bayesian model studied here is closely related to the LDA model, we also review existing approaches of incorporating supervised information into LDA training. We then present our proposed VDM model and describe a novel approach of deriving word priors using relative entropy from DBpedia<sup>1</sup> articles and tweets annotated using OpenCalais<sup>2</sup>. Following that, we present the dataset used in the paper and discuss experimental results obtained in comparison to a few baselines. Finally, we conclude the paper.

<sup>1</sup><http://dbpedia.org>

<sup>2</sup><http://www.opencalais.com>

## 2 Related Work

The task of detecting violent-related tweets can be viewed as a topical classification (TC) problem in which a tweet is labelled either as violent or non-violent related. Since the annotation of Twitter content is costly, some approaches have started to explore the incorporation of features extracted from external knowledge sources (KS) and the use of unsupervised or semi-supervised approaches to solve the TC problem. Since the model proposed in this paper makes use of both external KSs and topic models, we have divided the review of related work into approaches which rely on external KSs and approaches based on LDA model learning.

In the first case, Genc et al. (2011) proposed a latent semantic topic modelling approach, which mapped a tweet to the most similar Wikipedia<sup>3</sup> articles based on the tweets’ lexical features. Song et al. (2011) mapped a tweet’s terms to the most likely resources in the Probbase KS. These resources were used as additional features in a clustering algorithm which outperformed the simple bag of words approach. Munoz et al. (2011) proposed an unsupervised vector space model for assigning DBpedia URIs to tweets in Spanish. They used syntactical features derived from PoS (part-of-speech) tagging, extracting entities using the Sem4Tags tagger (Garcia-Silva et al., 2010) and assigning DBpedia URIs to those entities by considering the words appearing in the context of an entity inside the tweets. In contrast to these approaches, rather than labelling a tweet with KS URIs, we make use of DBpedia violence-related articles as one possible source of information from which prior lexicons can be derived.

Recently, Cano et al. (2013) proposed a supervised approach which makes use of the linked structure of multiple knowledge sources for the classification of Tweets, by incorporating semantic metagraphs into the feature space. However, in this study rather than extending the feature space with DBpedia derived features, we propose a strategy for characterising Violence related topics through the use of relative entropy, which filters out irrelevant word features. Moreover the proposed VDM model not only classifies documents as violent-related but also derives coherent category-topics (collection of words labelled as violent-related and non-violent related).

Our VDM model incorporates word prior knowledge into model learning. Here, we also review existing approaches for the incorporation of supervised information into LDA model learning. The supervised LDA (sLDA) (Blei and McAuliffe, 2008) uses empirical topic frequencies as a covariant for

<sup>3</sup><http://wikipedia.org>

a regression on document labels such as movie ratings. The Dirichlet-multinomial regression (DMR) model (Mimno and McCallum, 2008) uses a log-linear prior on document-topic distributions that is a function of observed meta data of the document. Labeled LDA (Ramage et al., 2009) defines a one-to-one correspondence between LDA’s latent topics and observed document labels and utilize a transformation matrix to modify Dirichlet priors. Partially Labeled LDA (PLDA) extends Labeled LDA to incorporate per-label latent topics (Ramage et al., 2011). The DF-LDA model (Andrzejewski et al., 2009) employs must-link and cannot-link constraints as Dirichlet Forest priors for LDA learning, but it suffers the scalability issue. Most recently, the aspect extraction model for sentiment analysis (Mukherjee and Liu, 2012) assumes that a seed set is given which consists of words together with their respective aspect category. Then depending on whether a word is a seed or non-seed word, a different route of multinomial distribution will be taken to emit the word. Our work was partially inspired by the previously proposed joint sentiment-topic model (JST) (Lin and He, 2009; Lin et al., 2012), which extracts topics grouped under different sentiments, relying only on domain-independent polarity word prior information.

While the afore-mentioned approaches assume the existence of either document label information or word prior knowledge, we propose to learn word prior knowledge using relative entropy from DBpedia and tweets annotated using OpenCalais. Moreover the proposed VDM model relies on the assumptions that the document category-topic distribution is shared across all documents in a corpus and words are generated either from a category-specific topic distribution or from a general background distribution. As we will discuss in section 5 these assumptions along with the proposed strategies for prior lexicon derivation show promising results outperforming various other topic models.

### 3 Violence Detection Model (VDM)

We propose a weakly-supervised violence detection model (VDM) here. In this model violence labels are associated with documents, under which topics are associated with violence labels and words are associated with both violence labels and topics. The graphical model of VDM is shown in Figure 1.

Assume a corpus of  $D$  documents denoted as  $\mathcal{D} = \{d_1, d_2, \dots, d_D\}$ ; where each document consists of a sequence of  $N_d$  words denoted by  $d = (w_1, w_2, \dots, w_{N_d})$ ; and each word in a document is an item from a vocabulary index of  $V$  different terms denoted by  $1, 2, \dots, V$ . We also assume that when an author writes a tweet message, she first decides whether

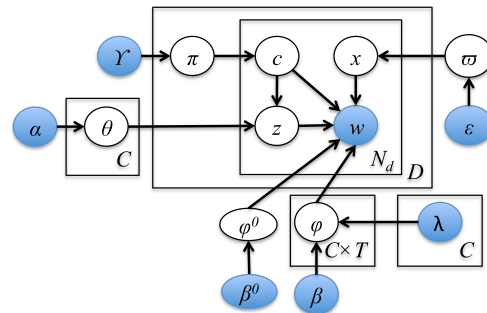


Figure 1: Violence detection model (VDM).

the tweet is violent-related or not. We use a category variable  $c$  to indicate violent-related topics or non-violent topics. If  $c = 0$ , the tweet is non-violent and the tweet topic is drawn from a general topic distribution  $\theta_0$ . If  $c = 1$ , the tweet is violent-related and the tweet topic is drawn from a violent category specific topic distribution  $\theta_1$ . Finally, each word of the tweet message is generated from either the background word distribution  $\phi^0$ , or the multinomial word distribution for the violent-related topics  $\phi_{c,z}$ . The generative process of VDM is shown below.

- Draw  $\omega \sim \text{Beta}(\epsilon), \varphi^0 \sim \text{Dirichlet}(\beta^0), \varphi \sim \text{Dirichlet}(\beta)$ .
- For each tweet category  $c = 1, \dots, C$ ,
  - for each topic  $z$  under the tweet category  $c$ , draw  $\theta_{cz} \sim \text{Dirichlet}(\alpha)$ .
- For each document  $m \in \{1..D\}$ ,
  - draw  $\pi_m \sim \text{Dirichlet}(\gamma)$ ,
  - For each word  $n \in \{1..N_d\}$  in document  $m$ ,
    - \* draw  $x_{m,n} \sim \text{Multinomial}(\omega)$ ;
    - \* if  $x_{m,n} = 0$ ,
      - draw a word  $w_{m,n} \sim \text{Multinomial}(\varphi^0)$ ;
    - \* if  $x_{m,n} = 1$ ,
      - draw a tweet category label  $c_{m,n} \sim \text{Multinomial}(\pi_m)$ ,
      - draw a topic  $z_{m,n} \sim \text{Multinomial}(\theta_{c_{m,n},n})$ ,
      - draw a word  $w_{m,n} \sim \text{Multinomial}(\varphi_{c_{m,n},z_{m,n}})$ .

We have a latent random variable  $x$  associated with each word token and acts as a switch. If  $x = 0$ , words are generated from a background distribution. If  $x = 1$ , words are sampled from the corpus-specific multinomial  $\varphi_{c,z}$  decided by the tweet category label (non-violent or violent)  $c$  and the tweet topic  $z$ .

#### 3.1 Model Inference

We use Collapsed Gibbs Sampling (Griffiths and Steyvers, 2004) to infer the parameters of the model and the latent violent categories and topics assignments for tweets, given observed data  $\mathcal{D}$ . Gibbs sampling is a Markov chain Monte Carlo method which

allows us to repeatedly sample from a Markov chain whose stationary distribution is the posterior of interest, switch variable  $x$ , category label  $c$ , and topic  $z$  here, from the distribution over that variable given the current values of all other variables and the data. Such samples can be used to empirically estimate the target distribution. Letting the index  $t = (m, n)$  denote  $n^{\text{th}}$  word in document  $m$  and the subscript  $-t$  denote a quantity that excludes data from  $n^{\text{th}}$  word position in document  $m$ , the conditional posterior for  $x_t$  is:

$$P(x_t = 0 | \mathbf{x}_{-t}, \mathbf{c}, \mathbf{z}, \mathbf{w}, \Lambda) \propto \frac{\{N_m^0\}_{-t} + \epsilon}{\{N_m\}_{-t} + 2\epsilon} \times \frac{\{N_{w_t}^0\}_{-t} + \beta^0}{\sum_{w'} \{N_{w'}\}_{-t} + V\beta^0}, \quad (1)$$

where  $N_m^0$  denotes the number of words in document  $m$  assigned to the background component,  $N_m$  is the total number of words in document  $m$ ,  $N_{w_t}^0$  is the number of times word  $w_t$  is sampled from the background distribution.

$$P(x_t = 1 | \mathbf{x}_{-t}, \mathbf{c}, \mathbf{z}, \mathbf{w}, \Lambda) \propto \frac{\{N_m^s\}_{-t} + \epsilon}{\{N_m\}_{-t} + 2\epsilon} \times \frac{\{N_{w_t}^s\}_{-t} + \beta}{\sum_{w'} \{N_{w'}\}_{-t} + V\beta}, \quad (2)$$

where  $N_m^s$  denotes the number of words in document  $m$  sampled from the category-topic distributions,  $N_{w_t}^s$  is the number of times word  $w_t$  is sampled from the category-topic specific distributions.

The conditional posterior for  $c_t$  and  $z_t$  is:

$$P(c_t = k, z_t = j | \mathbf{c}_{-t}, \mathbf{z}_{-t}, \mathbf{w}, \Lambda) \propto \frac{N_{d,k}^{-t} + \gamma}{N_d^{-t} + C\gamma} \cdot \frac{N_{d,k,j}^{-t} + \alpha_{k,j}}{N_{d,k}^{-t} + \sum_j \alpha_{k,j}} \cdot \frac{N_{k,j,w_t}^{-t} + \beta}{N_{k,j}^{-t} + V\beta}, \quad (3)$$

where  $N_{d,k}$  is the number of times category label  $k$  has been assigned to some word tokens in document  $d$ ,  $N_d$  is the total number of words in document  $d$ ,  $N_{d,k,j}$  is the number of times a word from document  $d$  has been associated with category label  $k$  and topic  $j$ ,  $N_{k,j,w_t}$  is the number of times word  $w_t$  appeared in topic  $j$  and with category label  $k$ , and  $N_{k,j}$  is the number of words assigned to topic  $j$  and category label  $k$ .

Once the assignments for all the latent variables are known, we can easily estimate the model parameters  $\{\pi, \theta, \varphi, \varphi^0, \omega\}$ . We set the symmetric prior  $\epsilon = 0.5$ ,  $\beta_0 = \beta = 0.01$ ,  $\gamma = (0.05 \times L)/C$ , where  $L$  is the average document length,  $C$  is the total number of category labels, and the value of 0.05 on average allocates 5% of probability mass for mixing. The asymmetric prior  $\alpha$  is learned directly from data using maximum-likelihood estimation (Minka, 2003)

and updated every 40 iterations during the Gibbs sampling procedure. We run Gibbs sampler for 1000 iterations and stop the iteration once the log-likelihood of the training data converges under the learned model.

### 3.2 Deriving Model Priors through Relative Entropy

Detecting violence and extremism from text closely relates to sentiment and affect analysis. While sentiment analysis primarily deals with positive, negative, or neutral polarities, affect analysis aims to map text to much richer emotion dimensions such as joy, sadness, anger, hate, disgust, fear, etc. In the same way violence analysis maps violence polarity into violence words such as looting, revolution, war, drugs and non-violent polarity to background words such as today, happy, afternoon. However, as opposed to sentiment and affect prior lexicon derivation, the generation of violence prior lexicons pose different challenges. While sentiment and affect lexicon, rarely changes in time, words relevant to violence tend to be event dependent.

In this section we introduce a novel approach for deriving word priors from social media, which is based on the measurement of the relative entropy of a word in a corpus. Assume a source corpus consisting of  $N$  documents denoted as  $\mathcal{SD} = \{\mathbf{sd}_1, \mathbf{sd}_2, \dots, \mathbf{sd}_N\}$ , where each document is labelled as not violent or violent. We define the following metrics:

1. **Corpus Word Entropy:** The entropy of word  $w$  in corpus  $\mathcal{SD}$  is measured as follows:

$$E_{SD}(w) = - \sum_{i=1}^N p(w | \mathbf{sd}_i) \log p(w | \mathbf{sd}_i), \quad (4)$$

where  $p(w | \mathbf{sd}_i)$  denotes the probability of word  $w$  given the document  $\mathbf{sd}_i$  and  $N$  the total number of documents.  $E_{SD}(w)$  captures the dispersion of the usage of word  $w$  in the corpus. Our intuition is that low entropy words are indicative of semantically coherent topics and therefore more informative, while high entropy words indicates words whose usage is more topical diverse and therefore less informative.

2. **Class Word Entropy:** The entropy of word  $w$  given the class label  $c$  is defined as follows:

$$E_{CWE}(w, c) = - \sum_{i=1}^N p(w | \mathbf{sd}_i^c) \log p(w | \mathbf{sd}_i^c), \quad (5)$$

where  $C$  denotes the number of classes (in our case violent and non-violent) and  $p(w | \mathbf{sd}_i^c)$  denotes the probability of word  $w$  given the document  $\mathbf{sd}_i$  in class  $c$ . In contrast to the general  $E_{SD}$ , the class word entropy characterises the usage of a word in a particular document class.

3. **Relative Word Entropy (RWE)**: In order to compare the word entropy used on documents in different categories, we measure the word relative entropy as follows:

$$RWE(w, c) = \frac{E_{CWE}(w, c)}{E_{SD}(w)} \quad (6)$$

The RWE provides information on the relative importance of that word to a given document class.

After deriving the RWE of each word given a class (i.e. violent or non-violent), we sorted words based on their RWE values in ascending order. Since our intuition is that lower entropy levels are more indicative of semantically coherent topics we choose the top  $K$  words of each class. We then built a matrix  $f$  of size  $K \times C$ , where  $C$  is the total number of document classes or category labels. The  $k$ th entry stores the probability that feature  $k$  is assigned with category label  $c$ . The matrix  $f$  essentially captures word prior knowledge and can be used to modify the Dirichlet prior  $\beta$  of category-topic-word distributions. We initialize each element of the matrix  $\beta$  of size  $C \times T \times V$  to 0.01 and then perform element-wise multiplication between  $\beta$  and  $f$  with the topic dimension ignored.

## 4 Experimental Setup

### 4.1 Dataset Description

The experimental setup consists of three stages: 1) derivation of word prior lexicon; 2) training of VDM and baselines; and 3) testing. For the first stage, we explored three different ways to construct a labelled document corpora for deriving prior lexicons. The first one is based on a Twitter corpus labelled using OpenCalais. This corpus comprises over 1 million tweets collected over a period of two months starting from November 2010. In order to build the Twitter-based violent dataset for deriving priors, we extracted tweets labelled as “War & Conflict” and considered them as violent annotations, while for the non-violent annotations we considered tweets annotated with labels other than this one (e.g. Education, Sports). We denote this dataset as **TW**. It is worth noting that the annotated results generated by OpenCalais are very noisy. We have evaluated OpenCalais on our manually annotated test set and only obtained an F-measure of 38%. Nevertheless, as will be seen later, word prior knowledge extracted from such noisy annotated tweets data is still very helpful in learning the VDM model for violence detection from tweets.

The second dataset for deriving priors is based on DBpedia which is a knowledge source derived from Wikipedia. The latest version of DBpedia consists of over 1.8 million resources, which have been classified

into 740 thousand Wikipedia categories, and over 18 million YAGO<sup>4</sup> categories. For constructing the violence related corpus we queried DBpedia for all articles belonging to categories and subcategories under the “violence” category, from which we kept their abstract as the document content. After removing those categories with less than 1000 articles, we obtained a set of 28 categories all related to violence. The resulting set of articles represented the violent set while for the non-violent rather than using non-violent related articles from DBpedia we opted for using the collection of Tweets from **TW** annotated as non-violent by OpenCalais. This decision was made in order to balance differences across the DBpedia and Twitter lexicons. This resulting dataset is referred to as **DB**.

Since the average word per article abstract in DBpedia exceeds the one of tweets, we decided to build a third dataset where the violent DBpedia documents resemble tweets in their size. In order to do so, we took into account that the average number of words per tweet in **TW** before preprocessing is 9.6. Then from each violent document in the **DB** dataset, we generated tweet size documents by chunking the abstracts into 9 or less words. We then combine the chunked documents from **DB** with **TW** and refer to the final dataset as **DCH**.

These datasets were used for deriving priors for the first stage. For the second stage, we built a training set of tweets derived from the TREC Microblog 2011 corpus<sup>5</sup>, which comprises over 16 million tweets sampled over a two week period (January 23rd to February 8th, 2011). This time period includes 49 different events including violence-related ones such as Egyptian revolution, and Moscow airport bombing, and non-violence related such as the Super Bowl seating fiasco. We sampled a subset of 10,581 tweets as our training set and manually annotated another 1,759 tweets as our test set. Details about the statistics of the training and testing datasets are presented in Table 1 under the label “Main Dataset”.

We preprocessed the described datasets by first removing: punctuation, numbers, non-alphabet characters, stop words, user mentions, links and hashtags. We then performed Lovins stemming in order to reduce the vocabulary size. Finally to address the issue of data sparseness, we removed words with a frequency lower than 5.

### 4.2 Deriving Model Priors

We derive word prior knowledge from the three datasets mentioned above, namely **TW**, **DB** and **DCH**; applying the relative word entropy (RWE)

<sup>4</sup><http://www.mpi-inf.mpg.de/yago-naga/yago/>

<sup>5</sup><http://trec.nist.gov/data/tweets/>

	Datasets for Priors		
	TW	DB	DCH
Vio	10,432	4,082	32,174
Non-Vio	11,411	11,411	11,411
	Main Dataset		
	Training Set	Testing Set	
Vio	10,581	759	
Non-Vio		1000	

Table 1: Document statistics of the datasets used for deriving prior lexicons and for training and testing the proposed model and baselines.

approach introduced in section 3.2 for word prior lexicon generation. For comparison purposes, we also employ the widely-used information gain (IG) method to select highly discriminative words under each class from the datasets. Table 2 presents the word statistics of the prior lexicons generated using these two different methods<sup>6</sup>. It worth noting that **DB** consists of 4,082 violent-related documents (DBpedia abstracts) and 11,411 non-violent documents (non-violent tweets). Since the average word per abstract is much larger in size than the one of a tweet, having a very low number of non-violent features selected using IG is expected as the violence class is over represented per violent document. This is the reason why we built another dataset by chunking the DBpedia abstracts to produce tweet-size documents (**DCH**). Having a balanced number of words per document in both violent and non-violent categories leads to more balanced priors, as shown in Table 2, where the number of non-violent features increased from 99 (in **DB**) to 1,345 (in **DCH**) using IG.

	IG			RWE		
	TW	DB	DCH	TW	DB	DCH
Vio	1,249	2,899	1,612	875	3,388	3,786
Non-Vio	1,749	99	1,345	2,595	879	2,438

Table 2: Statistics of the word prior lexicons.

### 4.3 Baselines

For comparison purposes, we have tested the following baselines:

**Learned from labelled features.** The word prior knowledge can be used as labelled feature constraints which can be incorporated into a MaxEnt classifier training with Generalized Expectation (GE) con-

<sup>6</sup>While the number of words selected for IG was set to 3000, the criteria for selecting the top  $K$  words in the RWE approach was based on taking the highest coherent level of entropy containing more than 5 words. Then from the sorted list of words we selected those whose entropy was smaller than this level.

straints (Druck et al., 2008) or Posterior Regularization (PR) (Ganchev et al., 2010). We use the implementation provided in MALLETT with default parameter configurations for our experiments and refer these two methods as *ME-GE* and *ME-PR* respectively.

**JST.** If we set the number of sentiment classes to 2 (violent or non-violent), then we can learn the Joint Sentiment-Topic (JST) model from data with the word prior knowledge incorporated in a similar way as the VDM model.

**PLDA.** The Partially-Labeled LDA (PLDA) (Ramage et al., 2011) model assumes that some document labels are observed and models per-label latent topics. It is somewhat similar to JST and VDM except that supervised information is incorporated at the document level rather than at the word level. The training set is labelled as violent or non-violent using OpenCalais. Such pseudo document labels are then incorporated into PLDA for training.

The hyperparameters of PLDA and JST are set to be the same as those for VDM.

## 5 Experimental Results

In this section we compare the overall classification performance of VDM and a set of proposed baselines. We performed a series of experiments to investigate the impact of the prior derivation strategies (RWE and IG) on classification performance, using the six prior lexicons introduced in Section 4.2. Some of the research questions addressed in this section are as follows: Do lexicons built from DBpedia contain useful features which can be applied for the violence classification of Tweets?; If so, to what extent these lexicons help the classification task?. We also present the overall evaluation of the proposed VDM against the proposed baselines based on the semantic coherence of the generated topics. All the experiments reported here were conducted using a 5 fold 3 trial setting.

### 5.1 Violence Classification Results vs. Different Word Priors

Table 3 compares the results obtained for violence classification for the proposed VDM model against the baselines, using prior lexicons derived with the proposed RWE strategy and the IG baseline approach. We can observe that although both *ME-GE* and *ME-PR* present a very high precision for word priors obtained from **TW** regardless using either IG or RWE, they also present a very low recall. This indicates that although the documents labelled as “violent” with these models were correctly identified, much of the rest of the violent documents in the testing set remained unidentified. We can also observe that the best results in terms of F-measure were obtained for the VDM model using the word priors derived from **TW** using RWE, which significantly outper-

	Prior	ME-GE			ME-PR			JST			VDM		
		<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
IG	TW	0.7737	0.0337	0.0646	0.6300	0.1034	0.1777	0.6939	0.9362	0.7969	0.75	0.9288	0.8297
	DB	0.4604	0.9704	0.6245	0.5634	0.6955	0.4773	0.6493	0.9228	0.7622●	0.6455	0.9141	0.7566
	DCH	0.4862	0.2447	0.3255	0.5680	0.2949	0.3274	0.7113	0.9291	0.8057	0.7575	0.92	0.8309
RWE	TW	0.7100	0.1342	0.2249	0.9125	0.0373	0.0717	0.7235	0.9296	0.8136	0.8258	0.8919	0.8575
	DB	0.4958	0.1844	0.2686	0.5303	0.0540	0.0981	0.6882	0.9421	0.7952	0.7024	0.9212	0.7969
	DCH	0.5161	0.1731	0.2588	0.8485	0.0091	0.0179	0.73	0.9351	0.8199	0.8189	0.8804	0.8484

Table 3: The performance of the classifiers using prior features derived from TW, DB and DCH ( $dbp + tw$ ). The number of topics is set to 5 for JST and VDM. The values highlighted in bold corresponds to the best results obtained in F-measure, while the shaded cells indicate the best results in F-measure for each scenario. Blank notes denotes that the F-measure of VDM significantly outperforms the baselines while ● denotes JST outperforms VDM. Significance levels:  $p$ -value  $< 0.01$

forms the baseline models ( $t$ -test with  $\alpha < 0.01$ ). To compare VDM against JST, we varied the topics  $T \in \{1, 5, 10, 15, 20, 25, 30\}$  and our significance test results revealed that VDM outperforms JST significantly ( $t$ -test with  $\alpha < 0.01$ ) over all the topic settings except for the JST using **DB** lexicon priors.

When comparing the effectiveness of the use of DBpedia as a source of prior lexicon, we can observe that the use of the full articles’ abstracts in the derivation of the prior lexicons **DB** did not present an improvement over the models based on Twitter derived lexicons (**TW**). However, the strategy of chunking DBpedia articles’ abstracts into tweet size documents (**DCH**), did help in boosting the overall F-measure in JST ( $t$ -test with  $\alpha < 0.05$ ). In the case of VDM, the use of **DCH** achieved an F-measure very close to the one obtained using Twitter prior lexicons (**TW**).

When comparing the effectiveness of the proposed RWE strategy against the IG baseline for deriving prior lexicons, we can observe that RWE consistently outperformed in F-measure for the JST and VDM models on all the three prior lexicon scenarios with the improvement ranging between 1-4% although it fails to boost F-measure on both ME-GE and ME-PR.

In the subsequent experiments, we incorporated word prior knowledge extracted from **TW** using our proposed RWE method.

## 5.2 Varying Number of Topics

We compare the violence classification accuracy of our proposed VDM model against PLDA and JST with different topic number settings. It can be observed from Figure 2 that with single topic setting, all the three models give a similar violence classification results. However, when increasing the number of topics, PLDA performs much worse than both JST and VDM with the violence classification accuracy stabilising around 60%. In PLDA, document labels of the training set were obtained using OpenCalais. As mentioned in Section 4.1, OpenCalais gave an F-measure of 38% for violence classification on the test

set. Hence document labels of the training set are not reliable. This explains the low classification accuracy of PLDA.

VDM gives fairly stable violence classification results across different topic numbers. The violence classification accuracy using JST attains the best with single topic and drops slightly with the increasing number of topics. This is because JST assumes the per-tweet category-topic distribution and potentially generates less coherent topics which affects the violence classification accuracy.

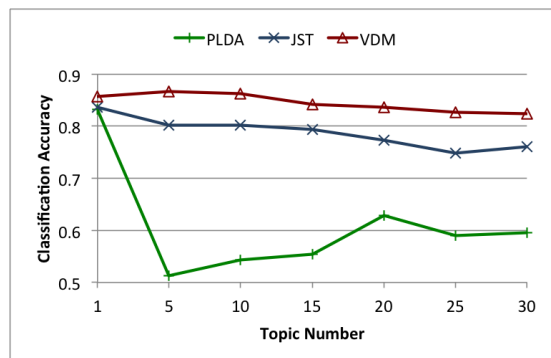


Figure 2: Violence classification Accuracy versus different topic numbers.

## 5.3 Topic Extraction

Table 4 presents two topic examples of violent and non-violent topics generated by VDM, JST and PLDA. We can observe that the topics revealed by VDM are representative of some of the events appearing during January/February 2011. For example, T1 gives an insight on the spreading of the Middle East Arab revolution, while T2 provides information regarding the Moscow airport bombing. For the case of non-violent topics, VDM revealed topics which appeared to be less semantically coherent than those of violent topics. However when reading the non-violent VDM T1, it gives an insight of the super bowl game related to the Jets. When checking the topics revealed



VDM				JST				PLDA			
Violent		Non-Violent		Violent		Non-Violent		Violent		Non-Violent	
T1	T2	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2
middle	crash	bowl	people	middle	crash	game	day	kill	game	crash	wow
east	kill	game	hate	government	nat	win	good	moscow	win	polic	cut
give	moscow	win	give	police	museum	jets	free	bomb	jets	drug	block
power	bomb	jets	damn	revolution	moscow	bowl	people	airport	watch	protester	arm
idea	airport	fan	shit	world	loot	fan	thing	leave	today	arrest	till
government	tweets	watch	miss	arm	report	reason	work	islam	play	car	officer
live	thought	today	fuck	streets	bomb	go	hope	injure	car	people	nat
time	injure	gone	hah	day	airport	damn	life	crash	fan	kill	fire
fall	arrest	damn	close	watch	kill	injure	today	report	damn	top	support
spread	dead	car	guy	live	morn	play	hah	victim	hate	part	london
upris	world	friends	sense	support	secure	run	back	terror	best	show	american

Table 4: Topic examples extracted under Violent and Non-Violent Labels for topic setting of 30 topics.

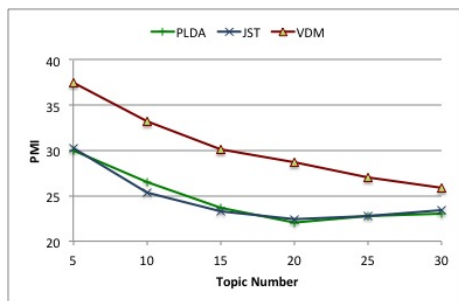
by JST, we can observe that although words seem to be semantically coherent for both violent and non-violent topics, there are words which belong to different violent events. For example the JST violent T2 mixes the Moscow bombing event with the Egyptian protesters Museum attack event. When checking the topics produced by PLDA we can see that it fails to correctly characterise violent and non-violent topics, since PLDA T2 should have been clearly classified as non-violent and the non-violent PLDA T1 as violent. Moreover in the violent PLDA T1 topic which presents violent related words, we can empirically identify more than one event involved.

In order to measure the semantic topical coherence of VDM and the proposed baselines, we made use of the Pointwise Mutual Information (PMI) metric proposed in (Newman et al., 2010). PMI is an automatic topic coherence evaluation which has been found to correspond well with human judgements on topic coherence. In particular, a coherent topic should only contain semantically related words and hence any pair of the top words from the same topic should have a large PMI value. For each topic, we compute its PMI by averaging over the PMI of all the word pairs extracted from the top 10 topic words. Figure 3 shows the PMI values of topics extracted under the violence and non-violence classes with the topic numbers varying between 5 and 30. It can be observed that JST and PLDA give similar PMI results. However, VDM outperforms both by a large margin.

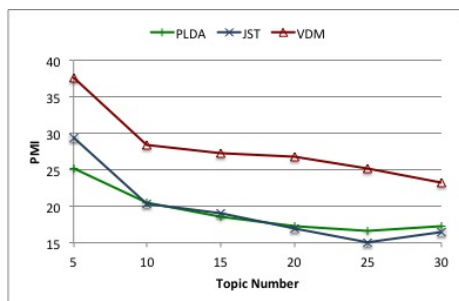
## 6 Conclusions and Future Work

In this paper, we have proposed a novel violence detection model (VDM), which enables the identification of text containing violent content and extraction of violence-related topics over social media data. VDM learning requires the incorporation of word prior knowledge which captures whether a word indicates violence or not. We propose a novel approach of deriving word prior knowledge using the measurement of relative entropy of words (RWE). Extensive experiments on the tweets data sampled from the TREC Microblog 2011 dataset show that our proposed RWE is more effective in deriving word prior knowledge compared to information gain. Moreover, the VDM model gives significantly better violence classification results compared to a few competitive baselines. It also extracts more coherent topics.

In future work, we intend to explore online learning strategies for VDM to adaptively update its parameters so that it can be used for violence detection from social streaming data in real-time.



(a) Violent topics.



(b) Non-violent topics.

Figure 3: Topic coherence measurement based on PMI. A larger PMI value indicates a better model.

## Acknowledgments

This work was partially supported by the EPSRC and DSTL under the grant EP/J020427/1 and the Visiting Fellowship funded by the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

## References

- David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML*, pages 25–32.
- D.M. Blei and J. McAuliffe. 2008. Supervised topic models. In *NIPS*, 20:121–128.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- A. E. Cano, A. Varga, F. Ciravegna, and Y. He. 2013. Harnessing linked knowledge source for topic classification in social media. In *Proceeding of the 24th ACM Conference on Hypertext and Social Media (Hypertext)*.
- G. Druck, G. Mann, and A. McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR*, pages 595–602.
- Evgeniy Gabilovich and Shaul Markovitch. 2006. Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. In *AAAI*.
- Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049.
- A. Garcia-Silva, Oscar Corcho, and J. Gracia. 2010. Associating semantics to multilingual tags in folksonomies.
- Yegin Genc, Yasuaki Sakamoto, and Jeffrey V. Nickerson. 2011. Discovering context: classifying tweets through a semantic transform based on wikipedia. In *Proceedings of the 6th international conference on Foundations of augmented cognition: directing the future of adaptive systems*, pages 484–492, Berlin, Heidelberg. Springer-Verlag.
- T.L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.
- Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. 2011. Emerging topic detection using dictionary learning. In *CIKM*, pages 745–754, New York, NY, USA. ACM.
- C. Lin and Y. He. 2009. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384.
- C. Lin, Y. He, R. Everson, and S. Rüger. 2012. Weakly-Supervised Joint Sentiment-Topic Detection from Text. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1134–1145.
- Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. 2012. Adding semantics to microblog posts. In *WSDM*, pages 563–572.
- Matthew Michelson and Sofus A. Macskassy. 2010. Discovering users’ topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, New York, NY, USA.
- D. Milne and I. H. Witten., editors. 2008. *Learning to link with Wikipedia*.
- D. Mimno and A. McCallum. 2008. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*.
- T. Minka. 2003. Estimating a Dirichlet distribution. Technical report.
- Óscar Muñoz García, Andrés García-Silva, Óscar Corcho, Manuel de la Higuera Hernández, and Carlos Navarro. 2011. Identifying Topics in Social Media Posts using DBpedia. In *Proceedings of the NEM Summit*, pages 81–86, September.
- Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *ACL*, pages 339–348.
- David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *NAACL*, pages 100–108.
- X. H. Phan, L. M. Nguyen, and S. Horiguchi. 2008. Learning to classify short and sparse text and web with hidden topics from large-scale data collections. In *WWW*.
- D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256.
- D. Ramage, C.D. Manning, and S. Dumais. 2011. Partially labeled topic models for interpretable text mining. In *KDD*.
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860.
- Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hong-song Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, pages 2330–2336.

# Detecting Spammers in Community Question Answering

Zhuoye Ding, Yeyun Gong, Yaqian Zhou, Qi Zhang, Xuanjing Huang

Fudan University

School of Computer Science

{09110240024,12110240006,zhouyaqian,qz,xjhuang}@fudan.edu.cn

## Abstract

As the popularity of Community Question Answering(CQA) increases, spamming activities also picked up in numbers and variety. On CQA sites, spammers often pretend to ask questions, and select answers which were published by their partners or themselves as the best answers. These fake best answers cannot be easily detected by neither existing methods nor common users. In this paper, we address the issue of detecting spammers on CQA sites. We formulate the task as an optimization problem. Social information is incorporated by adding graph regularization constraints to the text-based predictor. To evaluate the proposed approach, we crawled a data set from a CQA portal. Experimental results demonstrate that the proposed method can achieve better performance than some state-of-the-art methods.

## 1 Introduction

Due to the massive growth of Web 2.0 technologies, user-generated content has become a primary source of various types of content. Community Question Answering (CQA) services have also attracted continuously growing interest. They allow users to submit questions and answer questions asked by other users. A huge number of users contributed enormous questions and answers on popular CQA sites such as Yahoo! Answers<sup>1</sup>, Baidu Zhidao<sup>2</sup>, Facebook Questions<sup>3</sup>, and so on. According to a statistic from Yahoo, Yahoo! Answers receives more than 0.82 million questions

and answers per day<sup>4</sup>.

On CQA sites, users are primary contributors of content. The volunteer-driven mechanism brings many positive effects, including the rapid growth in size, great user experience, immediate response, and so on. However, the open access and reliance on users have also made these systems becoming targets of spammers. They post advertisements or other irrelevant answers aiming at spreading advertise or achieving other goals. Some spammers directly publish content to answer questions asked by common users. Additionally, another kind of spammers (we refer them as “*best answer spammers*”) create multiple user accounts, and use some accounts to ask a question, the others to provide answers which are selected as the best answers by themselves. They deliberately organize themselves in order to deceive readers. This kind of spammers are even more hazardous, since they are neither easily ignored nor identifiable by a human reader. Google Confucius CQA system also reported that best answer spammers may generate amounts of fake best answers, which could have a non-trivial impact on the quality of machine learning model (Si et al., 2010).

With the increasing requirements, spammer detection has received considerable attentions, including e-mails(L.Gomes et al., 2007; C.Wu et al., 2005), web spammer (Cheng et al., 2011), review spammer (Lim et al., 2010; N.Jindal and B.Liu, 2008; ott et al., 2011), social media spammer (Zhu et al., 2012; Bosma et al., 2012; Wang, 2010). However, little work has been done about spammers on CQA sites. Filling this need is a challenging task. The existing approaches of spam detection can be roughly into two directions. The first direction usually relied on costly human-labeled training data for building spam classifiers based on textual features (Y.Liu et al., 2008; Y.Xie et al.,

<sup>1</sup><http://answers.yahoo.com>

<sup>2</sup><http://zhidao.baidu.com>

<sup>3</sup><http://www.facebook.com>

<sup>4</sup><http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served>

2008; Ntoulas et al., 2006; Gyongyi and Molina, 2004). However, since fake best answers are well designed and lack of easily identifiable textual patterns, text-based methods cannot achieve satisfactory performance. Another direction relied solely on hyperlink graph in the web (Z.Gyongyi et al., 2004; Krishnan and Raj, 2006; Benczur et al., 2005). Although making good use of link information, link-based methods neglect the content-based information. Moreover, unlike the web, there is no explicit link structure on CQA sites. So two intuitive research questions are: (1) Is there any useful link-based structure for spammer detection in CQA? (2) If so, can the two techniques, i.e., content-based model and link-based model, be integrated together to complement each other for CQA spammer detection?

To address the problems, in this paper, we first investigate the link-based structure in CQA. Then we formulate the task as an optimization problem in the graph with an efficient solution. We learn a content-based predictor as an objective function. The link-based information is incorporated into textual predictor by the way of graph regularization. Finally, to evaluate the proposed approach, we crawled a large data set from a commercial CQA site. Experimental results demonstrate that our proposed method can improve the accuracy of spammer detection.

The major contributions of this work can be summarized as follows: (1) To the best of our knowledge, our work is the first study on spammer detection on CQA sites; (2) Our proposed optimization model can integrate the advantages of both content-based model and link-based model for CQA spammer detection. (3) Experimental results demonstrate that our method can improve accuracy of spammer detection.

The remaining of the paper is organized as follows: In section 2, we review a number of the state-of-the-art approaches in related areas. Section 3 analyzes the social network of CQA sites. Section 4 presents the proposed method. Experimental results in test collections and analysis are shown in section 5. Section 6 concludes this paper.

## 2 Related Work

Most of current studies on spam detection can be roughly divided into two categories: content-based model and link-based model.

Content-based method targets at extracting ev-

idences from textual descriptions of the content, treating the text corpus as a set of objects with associated attributes, and applying some classification methods to detect spam (P.Heymann et al., 2007; C.Castillo et al., 2007; Y.Liu et al., 2008; Y.Xie et al., 2008). Fetterly proposed quite a few statistical properties of web pages that could be used to detect content spam (D.Fetterly et al., 2004). Benevenuto went a step further by addressing the issue of detecting video spammers and promoters and applied the state-of-the-arts supervised classification algorithm to detect spammers and promoters (Benevenuto et al., 2009). Lee proposed and evaluated a honeypot-based approach for uncovering social spammers in online social systems (Lee et al., 2010). Wang proposed to improve spam classification on a microblogging platform (Wang, 2010).

An alternative web spam detection technique relies on link analysis algorithms, since a hyperlink often reflects some degree of similarity among pages (Gyongyi and Garcia-Molina, 2005; Gyongyi et al., 2006; Zhou et al., 2008). Corresponding algorithms include TrustRank (Z.Gyongyi et al., 2004) and AntiTrustRank (Krishnan and Raj, 2006), which used a seed set of Web pages with labels of trustiness or badness and propagate these labels through the link graph. Moreover, Benczur developed an algorithm called SpamRank which penalized suspicious pages when computing PageRank (Benczur et al., 2005).

## 3 Analysis on Social Network

Before analyzing the social network in CQA, we introduce some definitions. We refer users on CQA sites are someone who ask at least one question or answer at least one question. Moreover, users are divided into two categories: spammers and legitimate users. We define spammers as users who post at least one question or one answer intent to create spam.

A CQA site is particularly rich in user interactions. These interactions can be represented by Figure 1(a), where a particular question has a number of answers associated with it, represented by an edge from the question to each of the answer. We also include vertices representing authors of question or answers. An edge from a user to a question means that the user asked the question, and an edge from an answer to a user means that the answer was posted by this user. In the example,

a user  $U_1$  asks a question  $Q_1$ , while users  $U_4$ ,  $U_5$  and  $U_6$  answers this question. In order to observe the relation between users more clearly and directly, we summarize the relations between users as a graph shown in Figure 1(b). This graph contains vertices representing the users and omits the actual questions and answers that connect the users.

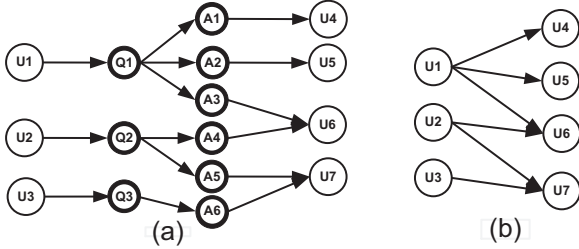


Figure 1: (a) Graph with users, questions, and answers in CQA; (b) Summary graph of users in CQA

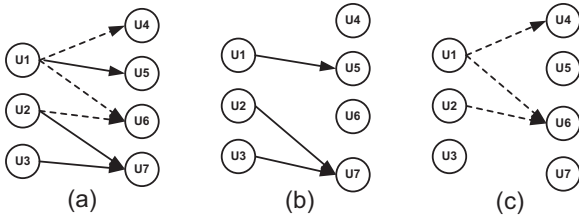


Figure 2: User graph with different relations in CQA (a) Question-answer relation; (b) Best-answer relation; (c) Non-best-answer relation

Three kinds of major relations among users on CQA sites are defined as follows:

**Question-answer relation:** As shown in Figure 2(a),  $U_4$  answers  $U_1$ 's question. We define that  $U_4$  and  $U_1$  have Question-answer relation. Furthermore, Question-answer relation can be divided into two disjoint sets: best-answer relation and non-best-answer relation.

**Best-answer relation:**  $U_1$  selects  $U_5$ 's answer as the best answer. We define that  $U_1$  and  $U_5$  have best-answer relation. The solid lines in Figure 2(b) express the best-answer relation.

**Non-best-answer relation:**  $U_1$  does not select  $U_4$ 's answer as the best answer. We define that  $U_1$  and  $U_4$  have non-best-answer relation. The dashed lines in Figure 2(c) express the non-best-answer relation.

### 3.1 Best-answer Consistency Property

From analyzing data crawled from CQA site, we present the following property about best-answer

relation:

**Best-answer consistency property:** If  $U_i$  selects  $U_j$ 's answer as the best answer, the classes of users  $U_i$  and  $U_j$  should be similar.

We explain this property as follows: consider that a legitimate user is unlikely to select a spammer's answer as the best answer due to its low quality, while a legitimate user is unlikely to answer a spammer's question, so the possibility of a spammer selecting a legitimate user's answer will also be small. This means that two users linked via best-answer relation are more likely to share similar property than two random users.

### 3.2 Characteristics of Best Answer Spammer

Different from the general spammers, some spammers generate many fake best answers to obtain higher status in the community. We refer them as *best answer spammers*. In order to generate fake best answers, a spammer creates multiple user accounts first. Then, it uses some of the accounts to ask questions, and others to provide answers. Such spammers may post low quality answers to their own questions, and select those as the best by themselves. They may generate lots of fake best answers, which may highly impact the user experience.

Furthermore, when the spammer's intention is just advertising, we can easily identify signs of its activity: repeated phone numbers or URLs and then ignore them. However, when the spammer's intention is to obtain higher reputation within the community, the spam content may lack obvious patterns. Fortunately, there are still some clues that may help identify best answer spammers. Two characteristics are described as follows:

**High best answer rate:** Best answer rate is the ratio of answers selected as the best answer among the total answers. This kind of spammers have an incredible high best answer rate, compared to normal users. Specifically, in a possible best answer spammer pair, sometimes only one user has an incredible high best answer rate. Because normally one responses for asking and another for answering. So we calculate the best answer rate  $BR(i, j)$  for a user pair  $(u_i, u_j)$  based on the maximum of their best answer rates:

$$BR(i, j) = \text{Max}(BR(i), BR(j)) \quad (1)$$

Where  $BR(i)$  is the best answer rate of  $u_i$ .

**Time margin score:** To be efficient, best answer spammers tend to answer their own ques-

tion quickly. We consider the time margin score  $Time(i, j)$  between a question posted and answered for  $u_i$  and  $u_j$  as an evidence.

$$Time(i, j) = \begin{cases} 1, & \text{if } TimeMargin(i, j) < \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where  $TimeMargin(i, j)$  is the real time margin between  $u_i$  asks a question and  $u_j$  answers this question and  $\varepsilon = 30$  minutes.

The best answer spammer score  $s(i, j)$  for a user pair  $(u_i, u_j)$  can be calculated as the combination of these two scores:

$$s(i, j) = \mu BR(i, j) + (1 - \mu)Time(i, j) \quad (3)$$

$\mu$  is trade-off of two scores, here we simply set  $\mu = 0.5$ . The value of  $s(i, j)$  is between 0 to 1. The higher  $s(i, j)$  is, the more likely  $u_i$  and  $u_j$  is a pair of the best answer spammers.

## 4 Spammer Detection on CQA Sites

In this section, the framework of our proposed approach is presented. First, the problem is formally defined. Next, we build a baseline supervised predictor that makes use of a variety of textual features, and then the consistency property and best answer spammer characteristics are incorporated by adding regularization to the textual predictor, last we discuss how to effectively optimize it.

### 4.1 Problem Statement

On CQA sites, there are three distinct types of entities: users  $U = \{u_1, \dots, u_{l+u}\}$ , answers  $A = \{a_1, \dots, a_M\}$ , and questions  $Q = \{q_1, \dots, q_N\}$ . The set of users  $U$  contains both  $U_L = \{u_1, \dots, u_l\}$  of  $l$  labeled users and  $U_U = \{u_{l+1}, \dots, u_{l+u}\}$  of  $u$  unlabeled users. We model the social network for  $U$  as a directed graph  $G = (U, E)$  with adjacency matrix  $A$ , where  $A_{ij} = 1$  if there is a link or edge from  $u_i$  to  $u_j$  and zero otherwise.

Given the input data  $\{U_L, U_U, G, Q, A\}$ , we want to learn a predictor  $c$  for a user  $u_i$ .

$$c(u_i) \rightarrow \{\text{spammer, legitimate user}\} \quad (4)$$

Legitimacy score  $y_i$  ( $0 \leq y_i \leq 1, i = 1, 2, \dots, n$ ) is computed for all the users. The lower  $y_i$  is, the more likely  $u_i$  is a spammer.

### 4.2 Text-based Spammer Prediction

In this subsection, we build a baseline predictor based on textual features in a supervised fashion.

We regard the legitimacy scores as generated by combining textual features.

We consider the following textual features.

- *The Length of answers*: The length may to some extent indicate the quality of the answer. The average length of answers is calculated as a feature.
- *The ratio of Ads words in answers*: Advertising of products is the main goal of a kind of spammers and they repeat some advertisement words in their answers.
- *The ratio of Ads words in questions*: Some spammers will refer some Ads in questions in order to get attention from more users.
- *The number of received answers*: The number of received answers can indicate the quality of the question.
- *Best answer rate*: Best answer rate can show the quality of their answers.
- *The number of answers*: It can indicate the authority of a user.
- *Relevance of question and answer*: We measure the average content similarity over a pair of question and answer which is computed using the standard cosine similarity over the bag-of-words vector representation.
- *Duplication of answers*: The Jaccard similarity of answers are applied to indicate the duplication of answers.

With these features, suppose there are in total  $k$  features for each user  $u_i$ , denoted as  $x_i$ . Then  $X = (x_1, x_2, \dots, x_n)$  is the  $k$ -by- $n$  feature matrix of all users. Based on these features, we define the legitimacy score of each user as follows,

$$y_i = w^T x_i \quad (5)$$

where  $w$  is a  $k$ -dimensional weight vector.

Suppose we have legitimate/spammer labels  $t_i$  in the training set.

$$t_i = \begin{cases} 1, & u_i \text{ is labeled as legitimate user} \\ 0, & u_i \text{ is labeled as spammer} \end{cases} \quad (6)$$

We will then define the loss term as follows,

$$\Omega(w) = \frac{1}{l} \sum_{i=1}^l (w^T x_i - t_i)^2 + \alpha w^T w \quad (7)$$

Once we have learned the weight vector  $w$ , we can apply it to any user feature vector and predict the class of unlabeled users.

### 4.3 Regularization for Consistency Property

In Section 4.2, each user is considered as a stand-alone item. In this subsection, we exploit social information to improve CQA spammer detection.

In Section 3.1, the consistency property has been analyzed that users connected via best-answer relation are more similar in property. So the property is enforced by adding a regularization term into the optimization model. The regularization is acted in a collection data set, including a small amount of labeled data ( $l$  users) and a large amount of unlabeled data ( $u$  users). Then the regularization term is formulated as:

$$REG_1(U) = \sum_{i,j}^{l+u} A_{ij} (y_i - y_j)^2 \quad (8)$$

Minimizing the regularization constraint will force users who have best-answer relation belong to the same class. We formulate this as graph regularization. The graph adjacency matrix  $A$  is defined as  $A_{ij} = 1$  if  $u_j$  selects  $u_i$ 's answer as the best answer, and zero otherwise. Then, Equation 8 becomes:

$$REG_1(w) = \sum_{i,j}^{l+u} A_{ij} (w^T x_i - w^T x_j)^2 \quad (9)$$

With this regularization, then the objective function Equation 7 becomes:

$$\begin{aligned} \Omega_1(w) = & \frac{1}{l} \sum_{i=1}^l (w^T x_i - t_i)^2 + \alpha w^T w \\ & + \beta \sum_{i,j}^{l+u} A_{ij} (w^T x_i - w^T x_j)^2 \end{aligned} \quad (10)$$

### 4.4 Regularization for Best Answer Spammer

In this subsection, we focus on best answer spammers. Since they cannot be easily detected by only textual features (Equation 7), we introduce an additional penalty score  $b_i$  to each user  $u_i$  which indicates the possibility of becoming a best answer

spammer. With the penalty score  $b_i$ , Equation 5 can be redefined as follows:

$$y_i = w^T x_i - b_i \quad (11)$$

where  $b_i$  is a non-negative score.

In order to obtain  $b_i$ , characteristics of best answer spammers are incorporated by adding graph regularization to the optimization problem. The regularization is also acted in a collection data set. Two kinds of regularization are presented as follows:

#### Penalty for Best Answer Spammers in Pairs

As described in Section 3.2, the score  $s(i, j)$  indicates the possibility of  $u_i$  and  $u_j$  becoming a pair of best answer spammers (Equation 3). We expect  $u_i$  and  $u_j$ , who create the spam together, should share this possibility together, as follows:  $b_i + b_j = e \times s(i, j)$ , where  $e$  is a penalty factor, we empirically set it to 0.5.

Then we can also formulate this as graph regularization as:

$$REG_2(b) = \sum_{i<j}^{l+u} A_{ij} (b_i + b_j - e \times s(i, j))^2 \quad (12)$$

#### Penalty Assignment for Individual User

After introducing a penalty score to the user pair  $(u_i, u_j)$ , we have to decide how they share this penalty.

Penalty is assigned to  $u_i$  and  $u_j$  similarly. This can be also formulated as graph regularization as follows:

$$REG_3(b) = \sum_{i<j}^{l+u} A_{ij} (b_i - b_j)^2 \quad (13)$$

With the regularization for best answer spammer, the objective function becomes:

$$\begin{aligned} \Omega_3(w, b) = & \frac{1}{l} \sum_{i=1}^l (w^T x_i - b_i - t_i)^2 + \alpha w^T w \\ & + \beta \sum_{i,j}^{l+u} A_{ij} ((w^T x_i - b_i) - (w^T x_j - b_j))^2 \\ & + \gamma \sum_{i<j}^{l+u} A_{ij} (b_i + b_j - e \times s(i, j))^2 \\ & + \delta \sum_{i<j}^{l+u} A_{ij} (b_i - b_j)^2 \end{aligned} \quad (14)$$

## 4.5 Optimization Problem

By considering all the components of the objective function introduced in the previous subsection, we can obtain the optimization problem. Our goal is to minimize the objective function to get optimal parameters vector  $w^*$  and penalty vector  $b$ . For solving the optimization problem, we apply a kind of limited-memory Quasi-Newton(LBFGS)(Liu and Nocedal, 1989). After obtaining the optimal parameter vector  $w^*$  and  $b$ , we can use the following scoring function  $y_i = w^{*T}x_i - b_i$  to calculate scores for unlabeled users. Users with low scores will be regarded as spammers.

## 5 Experiments

In this section, the experimental evaluation of our approach is presented. Firstly, we introduce the details of our data sets. Then the prediction performance of our proposed approach is compared with other methods. Finally, we test the contribution of the loss term and each regularization term on these real data sets and conduct some further analysis.

### 5.1 Data Collections

In order to evaluate our proposed approach to detect CQA spammers from the CQA site, we need a training/test collection of users, classified into the target categories. However, to the best of our knowledge, no such collection is currently available, thus requiring us to build one.

We consider a CQA user is a user if he has posted at least one question or one answer. Moreover, we define spammer as a user who intends to create one spam. Examples of spams are: (1) an advertisement of a product or web site. (2) Completely unrelated to the subject of question. A user that is not a spammer is considered legitimate. Then we will explain the strategy of crawling data from a CQA site, Baidu Zhidao, one of the most popular CQA site in China. We randomly select 50 seed users covering different topics, including sports, entertainment, medicine and technology. The crawler follows links of question asked and question answered, gathering information on different attributes of users, including content of all responded questions and answers. The crawler ran for one week, gathering 29,257 users and 299,815 Q&A pairs. From the collection data, we randomly select a training set of 1000 users for learning

process and a test set of 698 users for evaluation.

Three annotators were asked to label the users as spammers or legitimate users in both training and test set. All of the judges are Chinese and have used Baidu Zhidao frequently. The annotators judge the property of a user comprehensively based on the content information (quality of their answers, i.e. advertising and duplication of answers) and social information (interaction with other possible-spammers). The Cohen’s Kappa coefficient is around 0.85, showing fair to good agreement. And our test collection contains 698 users, including 525 legitimate users and 173 spammers.

### 5.2 Metrics and Settings

To measure the effectiveness of our proposed method, we use the standard metrics such as precision, recall, the F1 measure. Precision is the ratio of correctly predicted users among the total predicted users by system. Recall(R) is the ratio of correctly predicted users among the actual users manually assigned. F1 is a measure that trades off precision versus recall. F1 measure of the spammer class is  $2PR/(P + R)$ .

We fix the parameter  $\alpha$  in optimization method to 0.0005 which gives the best performance for the textual predictor and simply set the coefficients  $\beta = 0.5$   $\gamma = \delta = 1$  in the objective function. The problem of parameter sensitivity will be tested in Section 5.6. In the optimization process, initial value of  $w_i$  is set to a random value range from 0 to 1 and initial value of  $b_i$  is set to 0.

### 5.3 Comparison with Other Methods

Since there has been little work on QA spam detection, we implement four state-of-the-art methods for comparison, where TrustRank and AntiTrustRank are selected to represent link-based model, while Decision Tree and SVM are two content-based classifiers.

- **Our approach:** Optimization with regularization terms that Similarity with best-answer relation, penalty for Best answer spammer. (Equation 14)
- **TrustRank:** TrustRank is a well-known link-based method in Web spam detection, which is totally based on the Web link graph(Z.Gyongyi et al., 2004).



- **AntiTrustRank**: AntiTrustRank is another well-known link-based method, which assumes that a web page pointing to spam pages is likely to be spam (Krishnan and Raj, 2006).
- **Decision Tree**: Castillo et al. applied a base classifier, decision tree, for spam detection, the features include content-based and link-based features (C. Castillo et al., 2007).
- **SVM**: We applied another state-of-the-art classifier SVM (Cortes and Vapnik, 1995). The features are the same as that used in Decision Tree method.

Methods	Precision	Recall	F1
TrustRank	0.581	0.485	0.529
AntiTrustRank	0.632	0.545	0.585
Decision Tree	0.891	0.740	0.808
SVM	0.898	0.748	0.816
<b>Our approach</b>	<b>0.925</b>	<b>0.861</b>	<b>0.892</b>

Table 1: Performance comparison with other methods

In Table 1, the performance of each method is listed for comparison. From the table, we have the following observations.

First, taking the advantages of both content-based model and link-based model, our optimization approach outperforms baselines under all metrics. This indicates the robustness and effectiveness of our approach.

The second observation is link-based models (**TrustRank** and **AntiTrustRank**) cannot perform well. The explanations are as follows. (1) Link-based models rely solely on hyperlinks, without considering content-based features. However, as described in section 4.2, the content can provide a strong hint for detecting spammers. (2) A technical requirement of link-based model is that the link graph must be strongly connected, which may be the case in Web, but it is not the case in QA user question-answer graph. We measured on our collection dataset and found that the graph density (defined as  $D = \frac{2|E|}{|V|(|V|-1)}$  for a graph with vertices  $V$  and edges  $E$ ) of user question-answer graph is only  $10^{-4}$ . The small connectivity limits the performance of link-based model. This indicates that link-based models cannot be directly applied to CQA spammer detection. Considering

that our proposed approach can integrate content-based features and link-based features effectively, we regard our approach as very complementary to the state-of-the-art link-based methods.

Another observation is that the content-based classifiers underperform our approach. And **SVM** performs slightly better than **Decision Tree**. This shows the advantages of our proposed regularization in section 4. Regularization for consistency can propagate the labeled information among users, and regularization for best answer spammers help to identify the best answer spammers.

#### 5.4 Contribution of Loss and Regularization

In this subsection, we validate the contribution of our proposed loss term and regularization terms by the performance of real spammer detection task. And Table 2 lists the results of each method for comparison. We consider the following methods.

**BL**: Optimization using only content-based features. (Equation 7)

**REG:Sim**: Optimization with one regularization term that Similarity with best-answer relation. (Equation 10)

**REG:Sim+BAS**: Optimization with all regularization terms that Similarity with best-answer relation, penalty for Best Answer Spammer. (Equation 14)

Methods	Precision	Recall	F1
BL	0.911	0.711	0.798
REG:Sim	<b>0.945</b>	0.699	0.804
REG:Sim+BAS	0.925	0.861	<b>0.892</b>

Table 2: Performance of our optimization methods with different regularization for comparison

From the results we have the following observations: (1) Our content-based classifier **BL** performs well, due to the well-formed supervised learning model and reasonable features. (2) The performance of **REG:Sim** improves over **BL**, especially in the Precision measure because the social information is useful. (3) **REG:Sim+BAS** can significantly improve over **BL** especially in Recall measure. Because after adding penalty to best answer spammer, some best answer spammers can be detected successfully.

#### 5.5 Contribution of Content-based Features

In this subsection, we test the robustness of the features described in Section 4.2.

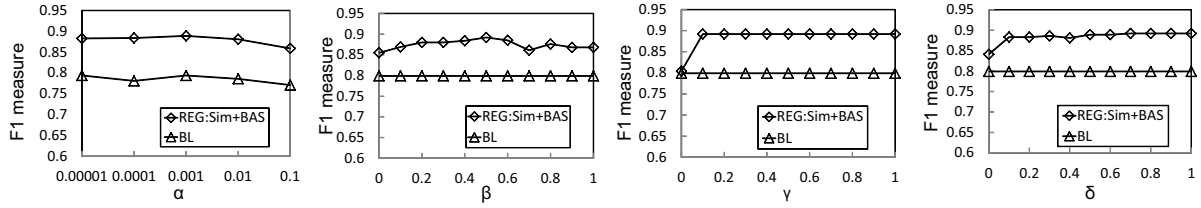


Figure 4: Parameter Sensitivity

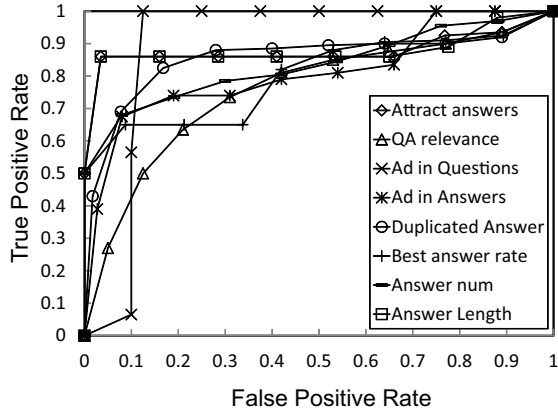


Figure 3: Content features comparison

To measure the discrimination power between spammers and legitimate users of each proposed attribute, we generate a Receiver Operating Characteristics (ROC) curve. ROC curves plot false positive rate on the X axis and true positive rate on the Y axis. The closer the ROC curve is to the upper left corner, the higher the overall accuracy is. Samples with the lowest scores (10%, 20%...100%) for each attribute are labeled as spammers respectively. The (ROC) curve are shown in Figure 3. Figure 3 shows the discrimination power of each content feature we described in Section 4.2. The first observation is that all of the content features are discriminative. The feature of Ads words in questions is the most powerful. Because few legitimate users will repeat Ads words in questions, so this feature can help to identify spammers more easily. Note that the feature of the best answer rate do not perform well. Because some best answer spammers also have high best answer rate.

## 5.6 Parameter Sensitivity

Our optimization approach have four parameters  $\alpha, \beta, \gamma, \delta$  to set: the tradeoff weight for each regularization term. The value of the regulariza-

tion weight controls our importance in the regularizer: a higher value results in a higher penalty when violating the corresponding regularization. So we mainly evaluate the sensitivity of our model with parameters by fixing all the other parameters and let one of  $\{\alpha, \beta, \gamma, \delta\}$  varies. Figure 4 shows the prediction performance in F1 measure varying each parameter. As we observed over a large range of parameters, our approach (**REG:Sim+BAS**) achieves significantly better performance than **BL** method. It indicates that the parameters selection will not critically affect the performance of our optimization approach.

## 6 Conclusion

In this paper, we first studied social networks on CQA sites. We found that spammers are usually connected to other spammers via the best-answer relation. We also studied the “best answer spammers” on CQA sites, which cannot be easily detected for lack of identifiable textual patterns. Our proposed model incorporated the link-based information by adding regularization constraints to the textual predictor. Experimental results demonstrated that our method is more effective for spammer detection compared to other state-of-the-art methods. Besides obtaining better performance, we have also analyzed the CQA social networks, which gives us insight on the model design.

## Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (61003092, 61073069), National Major Science and Technology Special Project of China (2014ZX03006005), Shanghai Municipal Science and Technology Commission (No.12511504500) and “Chen Guang” project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation(11CG05).

## References

- Andras A. Benczur, Karoly Csalogany, Tamas Sarlos, and Mate Uher. 2005. Spamrank-fully automatic link spam detection. In *AIRWeb'05*.
- Fabricio Benevenuto, Tiago Rodrigues, Virgilio Almeida, Jussara Almeida, and Marcos Goncalves. 2009. Detecting spammers and content promoters in online video social networks. In *Proceeding of SIGIR*.
- Maarten Bosma, Edgar Meij, and Wouter Weerkamp. 2012. A framework for unsupervised spam detection in social networking sites. In *Proceedings of ECIR*.
- C.Castillo, D.Donato, A.Gionis, V.Murdock, and F.Silvestri. 2007. Know your neighbors: Web spam detection using the web topology. In *Int'l ACM SIGIR*.
- Zhicong Cheng, Bin Gao, Congkai Sun, Yanbing Jiang, and Tie-Yan Liu. 2011. Let web spammers expose themselves. In *WSDM*.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20:273–297.
- C.Wu, K.Cheng, Q.Zhu, and Y.Wu. 2005. Using visual features for anti-spam filtering. In *IEEE Int'l Conference on Image Processing(ICIP)*.
- D.Fetterly, M.Manasse, and M.Najork. 2004. Spam,damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Int'l Workshop on the Web and Databases(WebDB)*.
- Zoltan Gyongyi and Hector Garcia-Molina. 2005. Link spam alliances. In *VLDB*.
- Zoltan Gyongyi and Hector Garcia Molina. 2004. Web spam taxonomy. Technical report, Stanford Digital Library Technologies Project.
- Zoltan Gyongyi, PavelBerkhin, Heter Garcia-Molina, and Jan O. Pedersen. 2006. Link spam detection based on mass estimation. In *VLDB*.
- Vijay Krishnan and Rashmi Raj. 2006. Web spam detection with anti-trust rank. In *ACM SIGIR workshop on adversarial information retrieval on the Web*.
- Kyumin Lee, James Caverlee, and Steve Webb. 2010. Uncovering social spammers: Social honeypots + machine learning. In *Proceeding of SIGIR*.
- L.Gomes, J.Almeida, V.Almeida, and W.Meira. 2007. Workload models of spam and legitimate e-mails. In *Performance Evaluation*.
- Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady W. Lauw. 2010. Detecting product review spammers using rating behaviors. In *CIKM*.
- Dong C. Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.
- N.Jindal and B.Liu. 2008. Opinion spam and analysis. In *WSDM*.
- Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In *Proceedings of WWW*.
- Myle ott, Yejin Choi, Claire Cardie, and Jeffrey T.Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *ACL*.
- P.Heymann, G.Koutrika, and H.Garcia-Molina. 2007. Fighting spam on social web sites: A survey of approaches and future challenges. In *IEEE Internet Computing*.
- Xiance Si, Edward Y. Chang, Zoltan Gyongyi, and Maosong Sun. 2010. Confucius and its intelligent disciples: integrating social with search. In *Proceeding of VLDB*.
- Alex Hai Wang. 2010. Don't follow me: Twitter spam detection. In *Proceedings of 5th International Conference on Security and Cryptography (SECRYPT)*.
- Y.Liu, H.Sundaram, Y.Chi, J.Tatemura, and B.Tseng. 2008. Detecting splogs via temporal dynamics using self-similarity analysis. In *ACM Transactions on the Web(TWeb)*.
- Y.Xie, F.Yu, K.Achan R.Panigrahy, G.Hulten, and I.Osipkov. 2008. Spamming botnet: Signatures and characteristics. In *ACM SIGCOMM*.
- Z.Gyongyi, H.Garcia-Molina, and J.pedersen. 2004. Combating web spam with trustrank. In *Int'l Conference on Very Large Data Bases(VLDB)*.
- Bin Zhou, Jian Pei, and ZhaoHui Tang. 2008. A spam-icity approach to web spam detection. In *SDM*.
- Yin Zhu, Xiao Wang, Erheng Zhong, Nanthan N. Liu, He Li, and Qiang Yang. 2012. Discovering spammers in social networks. In *Proceedings of AAAI*.

# Chinese Informal Word Normalization: an Experimental Study

Aobo Wang<sup>1\*</sup>, Min-Yen Kan<sup>1,2</sup>, Daniel Andrade<sup>3</sup>, Takashi Onishi<sup>3</sup>, Kai Ishikawa<sup>3</sup>

<sup>1</sup> Web IR / NLP Group (WING)

<sup>2</sup> Interactive and Digital Media Institute (IDMI)  
National University of Singapore

{wangaobo, kanmy}@comp.nus.edu.sg

<sup>3</sup> Knowledge Discovery Research Laboratories  
NEC Corporation, Nara, Japan

{s-andrade@cj, t-onishi@bq, k-ishikawa@dq}.jp.nec.com

## Abstract

We study the linguistic phenomenon of informal words in the domain of Chinese microtext and present a novel method for normalizing Chinese informal words to their formal equivalents. We formalize the task as a classification problem and propose rule-based and statistical features to model three plausible channels that explain the connection between formal and informal pairs. Our two-stage selection-classification model is evaluated on a crowdsourced corpus and achieves a normalization precision of 89.5% across the different channels, significantly improving the state-of-the-art.

## 1 Introduction

Microtext – including microblogs, comments, SMS, chat and instant messaging (collectively referred to as *microtext* by Gouwset *et al.* (2011) or *network informal language* by Xia *et al.* (2005)) – is receiving a larger research focus from the computational linguistic community. A key challenge is the presence of *informal words* – terms that manifest as *ad hoc* abbreviations, neologisms, unconventional spellings and phonetic substitutions. This phenomenon is so prevalent a challenge in Chinese microtext that the dual problems of informal word recognition and normalization deserve research. Given the close connection between an informal word and its formal equivalent, the restoration (normalization) of an informal word to its formal one is an important pre-processing step for NLP tasks that rely on string matching or word frequency statistics (Han *et al.*, 2012).

It is important to note that simply re-training models trained on formal text or annotated mi-

crotext is insufficient: user-generated microtexts exhibit markedly different orthographic and syntactic constraints compared to their formal equivalents. For example, consider the informal microtext “河蟹社会” (formally, “和谐社会”; “harmonious society”). A machine translation system may mistranslate it literally as “crab community” based on the meaning of its component words, if it lacks knowledge of the informal word “河蟹” (“和谐”; “harmonious”). It is thus desirable to normalize informal words to their standard formal equivalents before proceeding with standard text processing workflows.

In this work, we present a novel method for normalizing informal word to their formal equivalents. Specifically, given an informal word with its context as input, we generate hypotheses for its formal equivalents by searching the Google Web 1T corpus (Brants and Franz, 2006). Prospective informal-formal pairs are further classified by a supervised binary classifier to identify correct pairs. In the classification model, we incorporate both rule-based and statistical feature functions that are learned from both gold-standard annotation and formal domain synonym dictionaries. Also importantly, our method does not directly use words or lexica as features, keeping the learned model small yet robust to inevitable vocabulary change.

We evaluate our system on a crowdsourced corpus, achieving good performance with a normalization precision of 89.5%. We also show that the method can be effectively adapted to tackle the synonym acquisition task in the formal domain. To our best knowledge, this is the first work to systematically explore the informal word phenomenon in Chinese microtext. By using a formal domain corpus, we introduce a method that effectively normalizes Chinese informal words through different, independent channels.

\*This research is done in part during Aobo Wang’s internship in NEC Corporation.

## 2 Related Work

Previous works that address a similar task includes the study on abbreviations with their definitions (e.g., (Park and Byrd, 2001; Chang and Teng, 2006; Li and Yarowsky, 2008b)), abbreviations and acronyms in medical domain (Pakhomov, 2002), and transliteration (e.g., (Wu and Chang, 2007; Zhang et al., 2010; Bhargava and Kondrak, 2011)). These works dealt with such relations in formal text, but as we earlier argued, similar processing in the informal domain is quite different.

Probably the most related work to our method is Li and Yarowsky (2008a)’s work. They tackle the problem of identifying informal–formal Chinese word pairs in the Web domain. They employ the Baidu<sup>1</sup> search engine to obtain definition sentences – sentences that define or explain Chinese informal words with formal ones – from which the pairs are extracted and further ranked using a conditional log-linear model. Their method only works for definition sentences, where the assumption that the formal and informal equivalents co-occur nearby holds. However, this assumption does not hold in general social network microtext, as people often directly use informal words without any explanations or definitions.

While seminal, Li and Yarowsky’s method has other shortcomings. Relying on a search engine, the system recovers only highly frequent and conventional informal words that have been defined on the web, relying heavily on the quality of Baidu’s index. In addition, the features they proposed are limited to rule-based features and  $n$ -gram frequency, which does not permit their system to explain how the informal–formal word pair is related (*i.e.*, derived by which channel).

Normalizing informal words is another focus area in related work. An important channel for informal–formal mapping (as we review in detail later) is phonetic substitution. In work on Chinese, this is often done by measuring the Pinyin similarity<sup>2</sup> between an informal–formal pair. Li and Yarowsky (2008a) computed the Levenshtein distance ( $LD$ ) on the Pinyin of the two words in the pair to reflect the phonetic similarity. However, as a general string metric,  $LD$  does not

<sup>1</sup>[www.baidu.com](http://www.baidu.com)

<sup>2</sup>Pinyin is the official phonetic system for transcribing the sound of Chinese characters into Latin script.  $PYSim(x, y)$  is used to denote the similarity between two Pinyin string “ $x$ ” and “ $y$ ” hereafter.

capture the (dis-)similarity between two Pinyin pronunciations well as it is too coarse-grained. To overcome this shortcoming, Xia et al. (2008) propose a source channel model that is extended with phonetic mapping rules. They evaluated the model on manually-annotated phonetically similar informal–formal pairs. The disadvantage is that these rules need to be manually created and tuned. For example,  $Sim(chi, qi)$  is calculated as  $Sim(ch, q) * Sim(i, i)$  (here, “ $ch$ ” and “ $q$ ” are *Pinyin initials* and “ $i$ ” is a *Pinyin final*, as per convention), in which  $Sim(ch, q) = 0.8$  and  $Sim(i, i) = 1.0$  are defined manually by the annotators. As informal words and their usage in microtext continually evolve, they noted that it is difficult for annotators to accurately weigh the similarities for all pronunciation pairs. We concur that the labor of manually tuning weights is unnecessary, given annotated informal–formal pairs. Finally, we make the key observation that the similarity of initial and final pairs are not independent, but may vary contextually. As such, a decomposition of  $Sim(chi, qi)$  as  $Sim(ch, q) * Sim(i, i)$  may not be wholly accurate.

To tackle these problems as a whole, we propose a two-step solution to the normalization task, which involves formal candidate generation followed by candidate classification. Our pipeline relaxes the strong assumptions described by prior work and achieves significant improvement over the previous state-of-the-art.

## 3 Data Analysis

To bootstrap our work, we analyzed sample Chinese microtext, hoping to gain insight on how informal words relate to their formal counterparts. To do this, we first needed to compile a corpus of microtext and annotate them.

We utilized the Chinese social media archive, PrEV (Cui et al., 2012), to obtain Chinese microblog posts from the public timeline of Sina Weibo<sup>3</sup>, the most popular Chinese microtext site with over half a billion users. To assemble a corpus for annotation, we first followed the convention from (Wang et al., 2012) to preprocess and label *URLs*, *emoticons*, “*@usernames*” and *Hash-tags* as pre-defined words. We then employed *Zhubajie*<sup>4</sup>, one of China’s largest crowdsourcing platforms to obtain third-party (*i.e.*, not by the

<sup>3</sup><http://open.weibo.com>

<sup>4</sup><http://www.zhubajie.com>

original author of the microtext) annotations for any informal words, as well as their normalization, sentiment and motivation for its use (Wang et al., 2010). Our coarse-grained sentiment annotations use the three categories of “positive”, “neutral” and “negative”. Motivation is likewise annotated with the seven categories listed in Table 1:

to avoid (politically) <b>sensitive</b> words	17.8%
to be <b>humorous</b>	29.2%
to hedge criticism using <b>euphemisms</b>	12.1%
to be <b>terse</b>	25.4%
to <b>exaggerate</b> the post’s mood or emotion	10.5%
<b>others</b>	5.0%

Table 1: Categories used for motivation annotation, shown with their observed distribution.

In total, we spent US\$110 to annotate a subset of 5,500 posts (12,446 sentences), in which 1,658 unique informal words were annotated. Each post was annotated by three annotators where conflicts were resolved by simple majority. Annotations were completed after a five-week span and are publicly available<sup>5</sup> for comparative study.

### 3.1 Data Feature Analysis

From our observation of the annotated informal-formal word pairs, we identified three key channels through which the majority of informal words originate, summarized in Table 2. Here, the first column describes these channels, giving each channel’s observed frequency distribution as a percentage. Together, they account for about 94% of the channels by which informal words originate. The final “Motivation (%)” column also gives the distributional breakdown of motivations behind each of the channels as annotated by our crowdsourced annotators. We now discuss each channel.

**Phonetic Substitutions** form the most well-known channel where the resultant informal words are pronounced similar to their formal counterparts. It is also the channel responsible for most informal word derivation. It has been reported to account for 49.1% (Li and Yarowsky, 2008a) in the Web domain and for 99% in Chinese chats (Xia et al., 2006). In our study of the microtext domain, we found it to be responsible for 63% (Table 2). As highlighted in bold in the table, normalization in this channel is realized by a **character-**

<sup>5</sup><http://wing.comp.nus.edu.sg/portal/downloads.html>

**character** Pinyin mapping. An interesting special case occurs when the Chinese characters are substituted for Latin alphabets, where the alphabets form a Pinyin acronym. In these cases, each letter maps to a Pinyin initial (e.g., “bs” → ‘b’+ ‘s’ → “bi” + “shi” (鄙视(**bi shi**); “to despise”)), each of which maps to a single Chinese character. As such, we view this special case as also following the character-character mapping.

We found that phonetic substitutions are motivated by different intents. Slightly over half of the words are used to be humorous. This resonates well with the informal context of many microtexts, such that authors take advantage of expressing their humor through lexical choice. Another large group (28.9%) of informal words are variations of *politically sensitive words* (e.g., the names of politicians, religious movements and events), whose formal counterparts are often forbidden and censored by search engines or Chinese government officials. Netizens often create such phonetically equivalent or close variations to express themselves and communicate with others on such issues. An additional 18.7% of such word pairs are used euphemistically to avoid the usage of their harsher, formal equivalents. The remaining substitutions are explainable as typographical errors, transliterations, among other sources.

The **Abbreviation** channel contains informal words that are shortenings of formal words. Normalizing these informal words is equivalent to expanding short forms to corresponding full forms. As suggested by Chang and Teng (2006), we also agree that Chinese abbreviation expansion can be modeled as **character-word** mapping. The statistics in Table 2 suggest 19% of informal words come from this channel, and are used to save space and to make communication efficient, especially given the format and length limitations in microtext.

**Paraphrases** mark informal words that are created by a mixture of paraphrasing, abbreviating and combining existing formal words. We observe that the informal manifestation usually do not retain any of the original characters in their formal equivalents, but still retain the same meaning as a single formal word, or two meanings combined from two formal words. These words are created to enhance emotional response in an exaggerated (66.3%) and/or terse (27.3%) manner. For example in Table 2, “给力” as a whole comes from the

Channel (%)	Informal Word	Formal Word	Translation	Sentiment	Motivation (%)
<b>Phonetic Substitutions</b> (63)	河蟹(he2 xie4)	和谐(he2 xie2)	harmonious	positive	<b>sensitive</b> (28.9)
	鸭梨(ya1 li2)	压力(ya1 li4)	pressure	neutral	<b>humorous</b> (45.2)
	<b>bs</b>	鄙视(bi shi)	despise	negative	<b>euphemism</b> (18.7)
<b>Abbreviation</b> (19)	乘早(cheng2 zao3)	趁早(chen4 zao3)	as soon as possible	neutral	<b>others</b> (7.2)
	桌游 剧透	桌面_游戏 剧情_透露	board game tell the spoilers	neutral neutral	<b>terse</b> (100)
<b>Paraphrase</b> (12)	给力	很棒	awesome	positive	<b>exaggerate</b> (66.3)
	暴汗	非常_尴尬	very embarrassed	negative	<b>terse</b> (27.3)
	卖萌	可爱	cute	positive	<b>others</b> (6.4)

Table 2: Classification of Chinese informal words as originating from three primary channels. Pronunciation is indicated with Pinyin for phonetic substitutions, while characters in bold are linked to the motivation for the informal form.

paraphrase of the single formal word “很棒”, sharing the meaning of “awesome”. As another example, “暴汗” (“very embarrassed”) originates from two sources: “暴” meaning “十分” (“very”) and “汗” meaning “尴尬” (“embarrassed”). From this observation, we feel that both **character-word** and **word-word** mappings may adequately model the normalization process for this channel.

## 4 Methodology

Drawing on our observations, we propose a two step generation-classification model for informal word normalization. We first *generate* potential formal candidates for an input informal word by combing through the Google 1T corpus. This step is fast and generates a large, prospective set of candidates which are input to a second, subsequent classification. The subsequent classification is a binary yes/no classifier that takes both rule-based and statistical features derived from our identified three major channels to identify valid formal candidates.

Note that an informal word  $O$  (here,  $O$  for observation), even when used in a specific, windowed context  $C(O)$ , may have several different equivalent normalizations  $T$  (here,  $T$  for target). This occurs in the abbreviation (桌游 as (桌面 or 桌上) 游戏) and paraphrase (给力 很棒 or 很好 or 厉害) channels, where synonymous formal words are equivalent. In the case where an informal word is explainable as a phonetic substitution, only one formal form is viable. Our classification model caters for these multiple explanations.

Figure 1 illustrates the framework of the proposed approach. Given an input Chinese microblog post, we first segment the sentences into words and recognize informal words leveraging the approach proposed in (Wang and Kan, 2013).

For each recognized informal word  $O$ , we search the Chinese portion of the Google Web1T corpus using lexical patterns, obtaining  $n$  potential formal (normalized) candidates. Taking the informal word  $O$ , its occurrence context  $C(O)$ , and the formal candidate  $T$  together, we generate feature vectors for each three-tuple, i.e.,  $\langle O, C(O), T \rangle$ <sup>6</sup>, consisting of both rule-based and statistical features. These features are used in a supervised binary classifier to render the final yes (informal-informal pair) or no (not an appropriate formal word explanation for the given informal word) decision.

### 4.1 Pre-Processing

As an initial step, we can recognize informal words and segment the Chinese words in the sentence by applying joint inference based on a Factorial Conditional Random Field (FCRF) methodology (Wang and Kan, 2013). However, as our focus in this work is on the normalization task, we use the manually-annotated gold standard informal words ( $O$ ) and their formal equivalents ( $T$ ) provided in our annotated dataset. To derive the informal words’ context  $C(O)$ , we use the automatically-acquired output of the preprocessing FCRF, although noisy and a source of error.

### 4.2 Formal Candidate Generation

Given the two-tuple  $\langle O, C(O) \rangle$  generated from pre-processing, we produce a set of hypotheses  $|T|$  which are formal candidates corresponding to  $O$ . We use two assumptions to guide us in the selection of prospective formal equivalents of  $O$ . We first discuss Assumption 1 (as [A1]):

<sup>6</sup>For notational convenience, the informal word context  $C(O)$  is defined as  $W_{-i} \dots O \dots W_i$ ; here,  $i$  refers to the index of the word with respect to  $O$ , which we set in this work to 3.

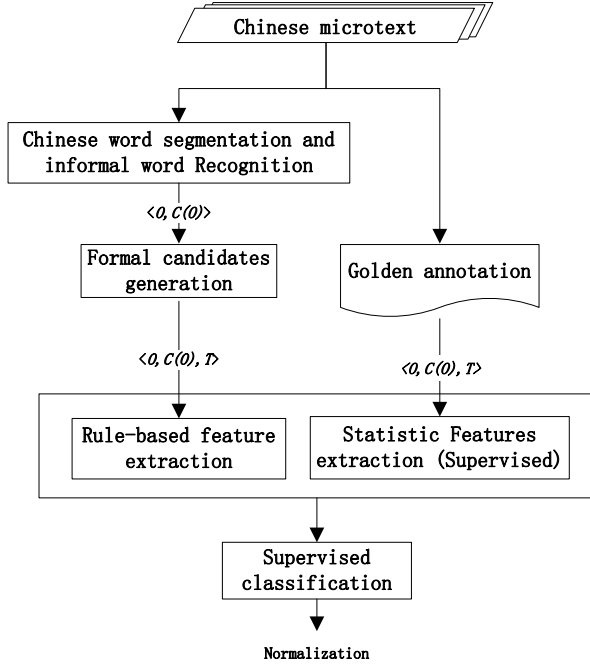


Figure 1: Our framework consists of the two steps of informal word recognition and normalization. Normalization breaks down to its component steps of candidate generation and classification.

[A1] The informal word and its formal equivalents share similar contextual collocations.

To implement [A1], we define several regular expression patterns to search the Chinese Web 1T corpus, as listed in Table 3. All entries that match at least one of the five rules are collected as formal candidates. Specifically,  $W_*$  refers to the word in context  $C(O)$ .  $T$  denotes any Chinese candidate word, and  $\hat{T}$  a word sharing at least one character in common with the informal word  $O$ .

$W_{-1} T W_1$	$W_{-2} W_{-1} T$	$T W_1 W_2$
$W_{-1} \hat{T}$		$\hat{T} W_1$

Table 3: Lexical patterns for candidate generation.

Our assumption is similar to the notion used for paraphrasing: that the informal version can be substituted for its formal equivalent(s), such that the original sentence’s semantics is preserved in the new sentence. For example, in the phrase “建设\_河蟹\_社会”, the informal word “河蟹” is exactly equivalent to its formal equivalent “和谐”, as the resulting phrase “建设\_和谐\_社会” (“build the harmonious society”) carries exactly the same semantics. This is inferrable when both the informal word  $O$  and the candidate share the same con-

textual collocations of “建设” and “社会”.

As the Web1T corpus consists of  $n$ -grams taken from approximately one trillion words indexed from Chinese web pages, queries for each informal word  $O$  can return long result lists of up to 20,000 candidates. To filter noise from the resulting candidates, we adopt Assumption 2 [A2]:

[A2] Both the original informal word in its context – as well as the substituted formal word within the same context – are frequent in the general domain.

We operationalize this by constraining the prospective normalization candidates to be within the top 1,000 candidates ranked by the trigram probability ( $P(W_{-1} T W_1)$ ). This probability is calculated by the BerkeleyLM (Pauls and Klein, 2011) trained over Google Web 1T corpus. Note that this constraint makes our method more efficient over a brute-force approach, in exchange for loss in recall. However, we feel that this trade-off is fair: by retaining the top 1000 candidates, we observed the loss rate of gold standard answers in each of the channels is 14%, 15%, and 17% for phonetic substitution, abbreviation and paraphrase, respectively. This is in comparison with the final loss rate of over 70% reported by Li and Yarowsky (2008a).

Given the annotations, the three-tuples ( $\langle O, C(O), T \rangle$ ) generated from the resulting list of candidates are labeled as  $Y$  ( $N$ ) as positive (negative) instances. As there are a much larger number of negative than positive instances for each  $O$ , this results in data skew.

### 4.3 Feature Extraction for Classification

For the classification step, we calculate both rule-based and statistical features for supervised machine learning. We leverage our previous observations to engineer features specific to a particular channel. We describe both classes of features, listing its type (*binary* or *continuous*) and which channel it models (*phonetic substitution*, *abbreviation*, *paraphrase*, or *all*), as a two tuple. We accompany each rule with an example, showing Pinyin and tones, when appropriate.

#### 4.3.1 Rule-based Features (5 features).

- $O$  contains valid Pinyin script  $\langle b, ph \rangle$   
e.g., “冻shi了” (“冻死si3了”; “too cold”)
- $O$  contains digits  $\langle b, ph \rangle$   
e.g., “v5” (“威wei1武wu3”; “mighty”)



- $O$  is a potential Pinyin acronym  $\langle b, ph \rangle$   
e.g., “bs” (“鄙bi3视shi4”; “despise”)
- $T$  contains characters in  $O$ ?  $\langle b, ph \rangle$   
e.g., “桌游” (“桌面游戏”; “board games”)
- The percentage of characters common between  $O$  and  $T$   $\langle c, all \rangle$

### 4.3.2 Statistical Features (7 features).

We describe these features in more detail, as they form a key contribution in this work. Note that the statistical features that leverage information from both informal and formal domains are derived via maximum likelihood estimation on the appropriate training data.

**Pinyin Similarity**  $\langle c, ph \rangle$ . Although Levenshtein distance ( $LD$ ; employed in (Li and Yarowsky, 2008a)) is a low cost metric to measure string similarity, it has its drawbacks when applied to Pinyin similarity. As an example, the informal word “淫yin2 才cai2” is normalized to “人ren2 才cai2”, meaning “talent”. This suggests that  $PYSim(yin, ren)$  should be high, as they compose an informal-formal pair. However this is in contrast to evidence given by  $LD$  as  $LD(yin, ren)$  is large (especially compared with the  $LD(yin, yi)$ , in which “yi” is a representative Pinyin string that has an edit distance with “yin” of just 1). For the manual annotation method, it is difficult for annotators to accurately weigh the similarities for all pronunciation pairs, since it is weighted arbitrarily. And the labor of manually tuning weights may be unnecessary, given annotated informal-formal pairs.

To tackle these drawbacks, we propose to fully utilize the gold standard annotation (i.e., informal-formal pairs applicable to the Phonetic Substitution channel) and to empirically estimate the Pinyin similarity from the corpus in a supervised manner. In our method, Pinyin similarity is formulated as:

$$PYSim(T|O) = \prod PYSim(t_i|o_i) \quad (1)$$

$$\begin{aligned} PYSim(t_i|o_i) &= PYSim(py(t_i)|py(o_i)) \\ &= \mu P(py(t_i)|py(o_i)) + \lambda P(ini(t_i)|py(o_i)) \\ &\quad + \eta P(fin(t_i)|py(o_i)) \end{aligned} \quad (2)$$

Here, the  $t_i$  ( $o_i$ ) stands for the  $i$ th character in word  $T$  ( $O$ ). Let the function  $py(x)$  return the

Pinyin string of a character and functions  $ini(x)$  ( $fin(x)$ ) return *initial* (*final*) of a Pinyin string  $x$ . We use linear interpolation algorithm for smoothing, with  $\mu$ ,  $\lambda$  and  $\eta$  as weights summing to unity. Then,  $P(py(t_i)|py(o_i))$ ,  $P(ini(t_i)|py(o_i))$  and  $P(fin(t_i)|py(o_i))$  are estimated using maximum likelihood estimation over the training set.

**Lexicon and Semantic Similarity**  $\langle c, ab + pa \rangle$ . For the remaining two channels, we extend the source channel model (SCM) (Brown et al., 1990) to estimate the character mapping probability. In our case, SCM aims to find the formal string  $T$  that the given input  $O$  is most likely normalized to.

$$\hat{T} = \arg \max_T P(T|O) = \arg \max_T P(O|T)P(T) \quad (3)$$

As discussed in Section 3, for both the two channels we use interpolation to model character-word mappings. Assuming the character-word mapping events are independent, we obtain:

$$P(O|T) = \prod P(o_i|t_i) \quad (4)$$

where  $o_i$  ( $t_i$ ) refers to  $i$ th character of  $O$  ( $T$ ). However, this SCM model suffers serious data sparsity problems, when the annotated microtext corpus is small (as in our case). To further address the sparsity, we extend the source channel model by inserting part-of-speech mapping models into Equation 4.

$$P(O|T) = \prod P'(o_i|t_i) \quad (5)$$

$$P'(o_i|t_i) = \alpha P(o_i|t_i) + \beta P(o_i|pos(t_i), pos(o_i)) \quad (6)$$

Here, let the function  $pos(x)$  return the part-of-speech (POS) tag of  $x$ <sup>7</sup>. Both  $P(o_i|t_i)$  and  $P(o_i|pos(t_i), pos(o_i))$  are then estimated using maximum likelihood estimation over the annotated corpus. In parallel with the Pinyin similarity estimation,  $\alpha$  and  $\beta$  are weights for the interpolation, summing to unity.

We give the intuition for our formulation.  $P(o_i|t_i)$  measures the probability of using character  $o_i$  to substitute for the given word  $t_i$ .  $P(o_i|pos(t_i), pos(o_i))$  measures the probability of using character  $o_i$  as the substitution of any word  $t_i$ , given the POS tag is mapped from  $pos(t_i)$  to  $pos(o_i)$ . Finally, given the limited availability of gold standard annotations, we can optionally use

<sup>7</sup>Implemented in our system by the FudanNLP toolkit <https://code.google.com/p/fudannlp/>.

formal domain synonym dictionaries to improve our model’s estimation lexical and semantic similarity.

**N-gram Probabilities**  $5 \times \langle c, all \rangle$ . We generate new sentences by substituting informal words with candidate formal words. The probabilities of the generated trigrams and bigrams (within a window size of 3) are computed with BerkeleyLM, trained on the Web1T corpus. The features capture how likely the candidate word is used in the informal domain. The five features are:

- Trigram probabilities:  $P(W_{-2}W_{-1}T)$ ;  $P(W_{-1}T W_1)$ ;  $P(T W_1 W_2)$
- Bigram probabilities:  $P(W_{-1} T)$ ;  $P(T W_1)$

## 5 Experiments

In our architecture, the candidate generation procedure is unsupervised. The part that does need tuning is the final, supervised classifier that renders the binary decision on each 3-tuple, as to whether the  $O-T$  pair is a match, so for this task we select the best classifier among three learners. The statistics reported by Li and Yarowsky (2008a) is then used as a baseline\* performance. We mark this with an asterisk to indicate that the comparison is just for reference, where the performance figures are taken directly from their published work, as we did not reimplement their method nor execute it on our contemporary data.

As a second analysis point, we compare our system – with and without features derived from synonym dictionaries – to assess how well our method adapts from formal corpora. Finally we show that our method is also effective to acquire synonyms for the formal domain (formal–formal pairs, in contrast to our task’s informal–formal pairs).

### 5.1 Data Preparation

We collected 1036 unique informal–formal word pairs with their informal contexts were collected from our annotated corpus for cross-fold validation. As any supervised classifier would do, we testing logistic regression (LR), support vector machine (SVM) and decision tree (DT) learning models, provided by WEKA3 (Hall et al., 2009). To acquire formal domain synonyms, we option-

ally employed the Cilin<sup>8</sup> and TYCDict<sup>9</sup> dictionaries.

## 5.2 Results

We adopt the standard metrics of precision, recall and  $F_1$  for the evaluation, focusing on the the positive (correctly matched as informal–formal pair)  $Y$  class.

### 5.2.1 Classifier choice

Table 4 presents the evaluation results over different classifiers. In this first experiment, data from all the channels are merged together and the result reported is the outcome of 5-fold cross validation. Lexicon similarity features are derived only from the training corpus. As the DT classifier performs best, we only report DT results for subsequent experiments.

Classifier	Pre	Rec	$F_1$
SVM	.646	.273	.383
LR	.567	.340	.430
DT (C4.5)	.886	.443	.590

Table 4: Performance comparison using different classifiers.

### 5.2.2 Comparison with Baseline\*

To make a direct comparison with the baseline\*, we perform cross-fold validation using data each of three channels separately. Since Li and Yarowsky (2008a) formalized the task as a ranking problem, we show the reported Top1 and Top10 precision in Table 5<sup>10</sup>.

Our model achieves high precision for each channel, compared with the baseline\* performance. From Table 5 we observe that normalizing words due to Phonetic Substitution is relatively easy as compared to the other two channels. That is because given the fixed vocabulary of standard Chinese Pinyin, the Pinyin similarity measured from the corpus is much more stable than the estimated lexicon or semantic similarity. The low recall for the Paraphrase channel suggests the difficulty of inferring the semantic similarity between word pairs.

<sup>8</sup>[http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE\\_user\\_op=view\\_page&PAGE\\_id=162](http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=162)

<sup>9</sup><http://www.datatang.com/data/29207/>

<sup>10</sup>Due to the difference in classification scheme, we re-computed the reported value, given our classification.

Channel	System	Pre	Rec	$F_1$
Phonetic Substitution	OurDT	.956	.822	.883
	LY Top1	.754	—	—
	LY Top10	.906	—	—
Abbreviation	OurDT	.807	.665	.729
	LY Top1	.118	—	—
	LY Top10	.412	—	—
Paraphrase	OurDT	.754	.331	.460
	LY Top1	—	—	—
	LY Top10	—	—	—

Table 5: Performance, analyzed per channel. “—” indicate no comparable prior reported results.

### 5.2.3 Final Loss Rate

We note that there is a tradeoff between the data scale and performance. By keeping the Top 1000 candidates, we observed an 18.8% overall loss of correct formal candidates (breaking down as 14.9% for Phonetic Substitutions, 22.8% for Abbreviations and 31.8% for Paraphrases). Based on this statistics, the final loss rate is 64.1%. By comparison, Li and Yarowsky (2008a)’s seed bootstrapped method’s self-stated loss rate is around 70%.

### 5.2.4 Channel Knowledge and Use of Formal Synonym Dictionaries

In the real-world, we have to infer the channel an informal word originates from. To assess how well our system does without channel knowledge, we merged the separate channel datasets together and train a single classifier.

To investigate the impact of the formal synonym dictionaries, two configurations – with and without features derived from synonym dictionaries – were also tested. To upper bound achievable performance, we trained an oracular model with the correct channel as an input feature. In the results presented in Table 6, we see that the introduction of the features from the formal synonym dictionaries enhances performance (especially recall) of the basic feature set. As upper-bound performance is still significantly higher, future work may aim to improve performance by first predicting the originating channel.

### 5.2.5 Formal Domain Synonym Acquisition

To evaluate our method in the formal text domain, we take the synonym pairs from TYCDict as the test corpus and use the microtext data together with Cilin dictionaries as training. The experiment

Feature set	Pre	Rec	$F_1$
w/o	.886	.443	.590
w	<b>.895</b>	<b>.583</b>	<b>.706</b>
w + channel	.915	.638	.752

Table 6: Performance over different feature sets. “w” (“w/o”) refers to the model trained with (without) features from formal synonym dictionaries. “channel” refers to the model trained with the correct channel given as an input feature.

follows the same workflow as is done for the earlier microtext experiments, except that the context is extracted from the Chinese Wikipedia<sup>11</sup>. As we obtained solid performance, ( $Pre = .949$ ,  $Rec = .554$  and  $F_1 = .699$ ), we feel that our method can be applied to synonym acquisition task in the formal domain.

## 6 Conclusion

Based on our observations from a crowdsourced annotated corpus of informal Chinese words, we perform a systematic analysis about how informal words originate. We show that there are three main channels – phonetic substitution, abbreviation and paraphrase – that are responsible for informal creation, and that the motivation for their creation varies by channel.

To operationalize informal word normalization we suggest a two-stage candidate generation-classification method. The results obtained are promising, bettering the current state of the art with respect to both  $F_1$  and loss rate. In our detailed analysis, we find that channel knowledge can still improve performance and is a possible field for future work.

## References

- Aditya Bhargava and Grzegorz Kondrak. 2011. How do you pronounce your name?: improving g2p with transliterations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 399–408.
- Thorsten Brants and Alex Franz. 2006. The google web 1t 5-gram corpus version 1.1. *LDC2006T13*.
- Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990.

<sup>11</sup>[http://en.wikipedia.org/wiki/Wikipedia:Database\\_download](http://en.wikipedia.org/wiki/Wikipedia:Database_download)

- A statistical approach to machine translation. *Computational linguistics*, pages 79–85.
- Jing-Shin Chang and Wei-Lun Teng. 2006. Mining atomic chinese abbreviations with a probabilistic single character recovery model. *Language Resources and Evaluation*, pages 367–374.
- A. Cui, L. Yang, D. Hou, M.Y. Kan, Y. Liu, M. Zhang, and S. Ma. 2012. PrEV: Preservation Explorer and Vault for Web 2.0 User-Generated Content. *Theory and Practice of Digital Libraries*, pages 101–112.
- Stephan Gouws, Donald Metzler, Congxing Cai, and Eduard Hovy. 2011. Contextual Bearing on Linguistic Variation in Social Media. In *Proceedings of the Workshop on Language in Social Media*, pages 20–29.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, pages 10–18.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012. 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.
- Z. Li and D. Yarowsky. 2008a. Mining and modeling relations between formal and informal Chinese phrases from web corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1031–1040.
- Zhifei Li and David Yarowsky. 2008b. Unsupervised translation induction for chinese abbreviations using monolingual corpora. In *Proceedings of ACL*, pages 425–433.
- Serguei Pakhomov. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 160–167.
- Youngja Park and Roy J Byrd. 2001. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 conference on empirical methods in natural language processing*, pages 126–133.
- Adam Pauls and Dan Klein. 2011. Faster and smaller n-gram language models. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 258–267.
- Aobo Wang and Min-Yen Kan. 2013. Mining informal language from chinese microtext: Joint word recognition and segmentation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–741.
- A. Wang, C.D.V. Hoang, and M.Y. Kan. 2010. Perspectives on crowdsourcing annotations for natural language processing, journal = Language Resources and Evaluation. pages 1–23.
- Aobo Wang, Tao Chen, and Min-Yen Kan. 2012. Retweeting From A Linguistic Perspective. In *Proceedings of the Second Workshop on Language in Social Media*, pages 46–55.
- K.F. Wong and Y. Xia. 2008. Normalization of Chinese Chat Language. *Language Resources and Evaluation*, pages 219–242.
- Jian-Cheng Wu and Jason S Chang. 2007. Learning to find english to chinese transliterations on the web. In *Proc. of EMNLP-CoNLL*, pages 996–1004.
- Y. Xia, K.F. Wong, and W. Gao. 2005. NIL Is Not Nothing: Recognition of Chinese Network Informal Language Expressions. In *4th SIGHAN Workshop on Chinese Language Processing*, volume 5.
- Yunqing Xia, Kam-Fai Wong, and Wenjie Li. 2006. A phonetic-based approach to chinese chat text normalization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 993–1000.
- Min Zhang, Xiangyu Duan, Vladimir Pervouchine, and Haizhou Li. 2010. Machine transliteration: leveraging on third languages. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1444–1452.

# Feature Selection Using a Semantic Hierarchy for Event Recognition and Type Classification

Yoonjae Jeong and Sung-Hyon Myaeng

Korea Advanced Institute of Science and Technology (KAIST)  
291 Daehak-ro (373-1 Guseong-dong), Yuseong-gu, Daejeon 305-701,  
Republic of Korea

{hybris, myaeng}@kaist.ac.kr

## Abstract

Event recognition and event type classification are among the important areas in text mining. A state-of-the-art approach utilizing deep-level lexical semantics and syntactic dependencies suffers from a limitation of requiring too large feature space. In this paper, we propose a novel feature selection method using a semantic hierarchy of features based on WordNet relations and syntactic dependencies. Compared to the well-known feature selection methods, our proposed method reduces the feature space significantly while keeping the same level of effectiveness. For noun events, it improves effectiveness as well as efficiency. Moreover, we expect the proposed feature selection can be applied to the other types of text classification using hierarchically organized semantic resources such as WordNet.

## 1 Introduction

Feature selection is an important issue in text-based classification because features can be generated in a number of different ways from text. Selecting features affects not only efficiency when the space is big but also classification effectiveness by eliminating noise features (Manning, Raghavan, & Schütze, 2008). In this paper, we propose a new feature selection method that utilizes semantic aspects of word features and discuss its relative merits compared to other well-known feature selection methods.

Among many text-based classification problems, this research focuses on event recognition

(a kind of binary classification) and type classification that have been studied extensively to improve performance of applications such as automatic summarization (Daniel, Radev, & Allison, 2003) and question answering (Pustejovsky, 2002). For event recognition and type classification, TimeML has served as a representative annotation scheme of events (Pustejovsky, Castaño, et al., 2003), which are defined as situations that happen or occur and expressed by verbs, nominalizations, adjectives, predicative clauses or prepositional phrases. TimeML defines seven types of events, REPORTING, PERCEPTION, ASPECTUAL, I\_ACTION, I\_STATE, STATE, and OCCURRENCE (Pustejovsky, Knippen, Littman, & Saurí, 2007), to which a recognized event text is classified for event type classification.

Different approaches to recognize and classify TimeML events have been proposed, ranging from rule-based approaches (Saurí, Knippen, Verhagen, & Pustejovsky, 2005) to supervised machine learning techniques based on lexical semantic classes and morpho-syntactic information around events (Bethard & Martin, 2006; Boguraev & Ando, 2007; Jeong & Myaeng, 2013; Llorens, Saquete, & Navarro-Colorado, 2010). Jeong & Myaeng (2013) recently showed that using the deeper-level of semantics increased the performance. They obtained the best performance in their classification experiments when lexical semantic features using hypernyms at the maximum depth of eight in WordNet were used for the event candidates and the words having syntactic dependency. While the approach showed a meaningful improvement, it has a problem of generating too many features.

Semantic features that can be mapped to a structure like WordNet have hierarchical relationships. In this situation, when two features

have a hypernym-hyponym relationship, the higher-level feature encompasses the lower-level one (see Figure 1-(a)). If a conventional feature selection method were used, therefore, the selected features would include both overly specific, low-level features and more general ancestors that cover the characteristics of the children (see Figure 1-(b)). When the general features are accurate and specific enough to represent the class, their descendants are unnecessary and redundant. When redundant features of similar kind are used, they cause not only efficiency problems but also potential overfitting of the model because the resulting model may become biased towards the semantics covered by the sub-tree containing the features.

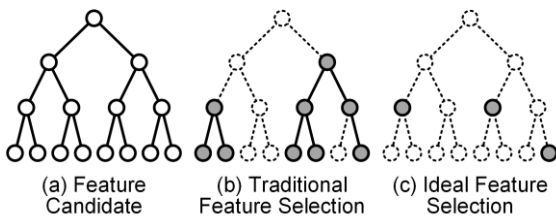


Figure 1. Feature Selection in Hierarchical Feature Space

It is important to select the features that are sufficiently general to encompass more specific features found in the training data but specific enough to utilize deep-level semantics available in the hierarchy (see Figure 1-(c)). The leftmost feature in (c) covers the semantics of the two features under it without having to keep them. Choosing the feature in the center and the rightmost feature has a similar effect and at the same time avoids using the overly general feature that encompasses both as well as the sibling of the rightmost one, which is not an appropriate one. In other words, we should select as general a feature as possible as long as none of them are considered irrelevant for the class, thereby it can cover the semantics of the features underneath it, without which we can achieve better efficiency.

In short, we propose a method for solving the problem of using features that are semantically redundant. Assuming that all the features can be organized in the form of a hierarchy, the method attempts to select the features that are as specific as possible as long as there are no semantically redundant features.

## 2 Event Recognition and Type Classification Task

We first describe the task for recognition and type classification of TimeML events. For word-

based event recognition and type classification, we converted the phrase-based annotations into a form with BIO-tags. For each word in a document, we assign a label indicating whether it is inside or out-side of an event (i.e., BIO2<sup>1</sup> label) as well as its type. For type classification, in addition, each word must be classified into one of the known event classes. Figure 2 illustrates an example of chunking and labeling components of an event in a sentence.

Word	Event Label	Event Type Label
All	O	O
75	O	O
people	O	O
on	B-EVENT	B-STATE
board	I-EVENT	I-STATE
the	O	O
Aeroflot	O	O
Airbus	O	O
died	B-EVENT	B-OCCURRENCE
.	O	O

Figure 2. Event chunking for a sentence, “All 75 people on board the Aeroflot Airbus died.” B-EVENT, I-EVENT and O refer to the beginning, inside and outside of an event.

Our method consists of three parts: preprocessing, feature extraction and selection, and classification. The preprocessing part analyzes raw text for tokenization, PoS tagging, and syntactic parsing (dependency parsing). It is done by the Stanford CoreNLP package<sup>2</sup>, which is a suite of natural language processing tools. Then, the feature extraction part converts the preprocessed data into the feature space, followed by feature selection. Finally, the classification part determines whether the given word is an event or not and its type using a maximum entropy (ME) classifier.

## 3 Feature Candidate Generation

Because the goal of the proposed method is to automatically select the most valuable features, we generate feature sets based on the same criteria of Jeong & Myaeng’s work (2013), which showed better performance for TimeML event than the state-of-the-art approach. The details are below:

<sup>1</sup> IOB2 format: (B)egin, (I)inside, and (O)utside

<sup>2</sup> Stanford CoreNLP, <http://nlp.stanford.edu/software/corenlp.shtml>

**Lexical Semantic Features (LSF).** The set of target words’ lemmas and their all-depth WordNet semantic classes (i.e., hypernyms). For example, a noun “*drop*” that is mapped to such a WordNet class is always an event regardless of its context in a sentence in the TimeBank corpus (Pustejovsky, Hanks, et al., 2003).

**Windows Features (WF).** The lemma, hypernyms, and PoS of the context defined by a five-word window [-2, +2] around a target word.

**Dependency-based Features (DF).** They are similar with WF, but the context is defined by syntactic dependencies. This feature type differs from WF because the context may go beyond the fixed size window and the features are not just words. Increasing the window size for WF instead of using this feature type is not an option because it would end up including some noise by including too big a context. Four dependencies we consider are: subject (SUBJ), object (OBJ), complement (COMP), and modifier (MOD).

- **SUBJ type.** A feature is formed with the governor or dependent word and its hypernyms that has the SUBJECT relation (*nsubj* and *nsubjpass*) with the target word.
- **OBJ type.** It is the governor or dependent word and its hypernyms, which has the OBJECT relation (*dobj*, *iobj*, and *pobj*) with the target word. In “... *delayed the game* ...”, for instance, the verb “*delay*” can describe the temporal state of its object noun, “*game*”.
- **COMP type.** It indicates the governor or dependent word and its hypernyms, which has the COMPLEMENT relation (*acomp* and *xcomp*) with the target word. In “... *called President Bush a liar* ...”, for example, the verb “*called*” makes the state of its object (“*Bush*”) into the complement noun, “*liar*”. In this case, the word “*liar*” becomes a STATE event.
- **MOD type.** It refers to the dependent words and their hypernyms in MODIFIER relation (*amod*, *advmod*, *partmod*, *tmod* and so on). This feature type is based on the intuition that some modifiers such as temporal expression reveal the word it modifies has a temporal state and therefore is likely to be an event.

**Combined Features (CF).** They are a combination of LSF and DF (or WF). A certain DF may not be an absolute clue for an event by itself but only when it co-occurs with a certain lexical or semantic aspect of the target word.

#### 4 Feature Selection Based on Semantic Hierarchy

Since a large number of features are generated with the aforementioned feature generation method, it is necessary to filter out those whose roles in classification are minimal. We first remove the feature candidates whose frequency in the training data is less than two. If a target word containing the feature candidate is determined not to be an event more than 50% in the training data, it is also eliminated. The remaining feature candidates are then organized into a meaning hierarchy so that we can apply the tree-based feature selection method.

An entailment relationship between two features,  $f_i \gg f_j$ , is established by a hypernym/hyponym relationship, syntactic dependency, or occurrence sequence as in Table 1. A and D represent an ancestor and a descendent in a feature hierarchy tree with  $A \gg D$ . We call the LSF and DF (or WF) features in CF as target and context elements, respectively. LSF can be an ancestor of CF because LSF does not consider the surrounding context of a target word whereas CF includes the context.  $CF_{LD}$  and  $CF_{LW}$  mean CF of LSF and DF and CF of LSF and WF, respectively.

A	D	Condition
LSF	LSF	A is hypernym of D.
LSF	$CF_{LD}$ or $CF_{LW}$	A is synset/hyponym of target in D. e.g.) $process_{LSF} \gg (report_{DF}, process_{LSF})$
DF	DF	Same dependency type. A is the hypernym of D.
DF	$CF_{LD}$	Same dependency with the target. A is synset/hyponym of surrounding in D. e.g.) $report_{DF} \gg (report_{DF}, process_{LSF})$
WF	WF	Same position from target. A is the hypernym of D.
WF	$CF_{LW}$	Same position from target. A is the synset/hyponym of the context in D. e.g.) $before_{WF} \gg (before_{WF}, launch_{LSF})$
$CF_{LD}$	$CF_{LD}$	Same dependency with target. The target and the context of A

A	D	Condition
		are the synset/hypernym of those of D, respectively.
CF <sub>LOW</sub>	CF <sub>LOW</sub>	Same position from target. The target and the context of A are the synset or hypernym of those of D, respectively.

A: Ancestor, D: Descendant (A >> D)

Table 1. Entailment Relation of Features

#### 4.1 Feature Tree Generation

Given that the entailment relationship >> can be established between two features, we can construct a feature tree that becomes a basis for tree-based feature selection. We begin with a tree that only has a root node R, a meta-feature that is the ancestor of all features. R entails and keeps adding new features to the tree until all the features are added to the tree. We define  $a$ ,  $d$ , and  $c$  for ancestor, descendent, and child features with the relationships  $a \gg d$  and  $a > c$  where  $>$  means  $c$  is a child of  $a$ , restricting that there is no node between  $a$  and  $c$  with  $a \gg c$ . Figure 3 illustrates the detail algorithm of feature tree generation.

When a new feature  $f$  is added to the (sub-)tree whose root is  $a$  and  $a \gg f$ ,  $f$  either becomes a child of  $a$  or is added to one of the sub-trees of  $a$  (line 9~28). If there is  $c$  such that  $c \gg f$ ,  $f$  is added to a subtree whose root is  $c$  (line 14~17). On the other hand, if  $f \gg c$ ,  $f$  replaces  $c$ , and  $c$  is entered to the sub-tree whose root is  $f$  (line 19~25). Finally if  $f$  has no entailment relation with any of the children nodes of  $a$ ,  $f$  is added as a child of  $a$  (line 26~27).

```

1 program GenerateTree;
2 F := feature candidates set;
3 r := root of feature tree;
4 begin
5   for f in F do
6     add_feature(r, f);
7   end;
8
9   procedure add_feature (a, f)
10  a : ascendant feature;
11  f : new feature;
12  begin
13    for c of a's children
14      if f is descendant of c then
15        begin
16          add_feature (c, f);
17          break;
18        end
19      else
20        if c is descendant of f then
21          begin
22            remove c from a's children;
23            add f to a's children;
24            add_feature (f, c);

```

```

24         break;
25       end;
26     if no child of a is ancestor or
27     descendant of f then
28       a's children <- f;
29     end.

```

Figure 3. Feature Tree Generation Algorithm

#### 4.2 Tree-Based Feature Selection

The key idea of the selection algorithm we devised is to evaluate each of the paths in the tree and select the appropriate node (i.e. feature). A path is defined to be the list of nodes between the root and a leaf node. In essence, the problem of selecting nodes or features from a tree is converted into smaller problems of selecting a node from individual paths. The process is illustrated with Figure 4 where each node of the tree except the root represents a feature. The tree has  $n$  paths corresponding to the number of leaf nodes. The algorithm selects the most representative node on a path, which is marked with a black node in Figure 4.

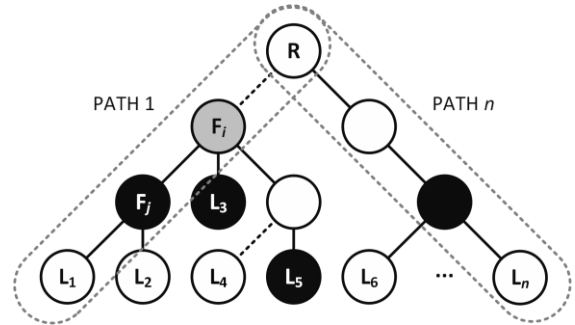


Figure 4. Paths between the root and the leaf nodes in a feature tree

To select the most representative feature on a path, we employed the notion of *lift*, which has been used widely in the area of association rule mining to compute the degree to which two items are associated (Tufféry, 2011). More specifically, it is defined as Equation (1) where  $P(f)$  indicate the probability of a feature  $f$  in training data set.  $P(E|f)$  is the conditional probability of events occurring given that  $f$  occurs.

$$lift(f) = \frac{P(E|f)}{P(f)} \quad (1)$$

While general feature selection methods such as  $\chi^2$  are based on the degree of belief, our selection method considers the reliability and applicability (or generality) of a feature. In other words, a feature we choose should have a high *lift* value (i.e., high reliability) and lie closest to the root on a path so that we can broaden its applicability.



These criteria would be particularly true when the amount of training data is not sufficient.

However, selecting the feature at the highest level in the tree may not be the best choice. In Figure 4, for example, even if the node  $F_i$  in grey is determined to be the most representative one for the path 1, it may not be the best one. In this case,  $F_j$  may be a better one because it happens to be the representative node for the path between  $F_i$  and  $L_1$ . However, there is a chance that the sub-tree of  $F_i$  may have important features (i.e.,  $L_3, L_5$ ) that end up elevating  $F_i$ 's weight unfairly. Instead of  $F_i$ , using  $F_j$  would be a better choice.

In order to handle this problem, we developed an algorithm where the key idea works as in Figure 5. We first collect all the representative features from the paths based on the reliability and generality criteria mentioned above (line 29~45). For each representative node, we check if any of the descendant nodes have been selected as a representative node of other paths (line 21). If the condition is met, the node is no longer considered as a representative node (line 23). The same process is applied to the sub-tree whose root is the node just deleted from the set of representative nodes (line 25). Up to now, this process does not require manually checking the performance for the selected features.

```

1  program SelectFeatures;
2  T := feature tree
3  F := selected feature set
4  begin
5    F ← ∅;
6    select_features(T);
7  end;
8
9  procedure select_from_tree
10 t := subtree of T
11 begin
12   r ← root of t
13   L ← leaf features of t
14   for l of L
15     p ← path from r to l;
16     f ← select_from_path (p);
17     add f to F;
18   end;
19   for f of F
20     D ← descendants of f;
21     if {d | d ∈ D and d ∈ F} ≠ ∅
22   then
23     begin
24       remove f from F;
25       t_f ← subtree of t whose root
26       is f
27       select_from_tree (t_f);
28     end;
29   end.
30 procedure select_from_path
31 p := feature path
32 begin

```

```

32   cur ← front of p
33   while
34     next ← next of cur
35     if next is null then
36   begin
37     add cur to F;
38     return;
39   end;
40   if lift(cur) ≥ lift(next) then
41   begin
42     add cur to F;
43     return;
44   end;
45 end.

```

Figure 5. Tree-Based Feature Selection Algorithm

We select the final features among those obtained through the above process by employing a widely used feature selection method (in our case,  $\chi^2$ ). It is because the most representative feature in a path might not be effective one in the entire feature space.

## 5 Experiment

### 5.1 Experimental Setup

The main goal of the experiment is to examine the efficacy of the proposed tree-based feature selection method in the context of event recognition and event type classification. For test collection, we use the TimeBank 1.2 corpus (Pustejovsky, Hanks, et al., 2003), which is the most recent version of TimeBank, annotated with the TimeML 1.2.1 specification. It contains 183 news articles and more than 61,000 non-punctuation tokens, among which 7,935 represent events.

We analyzed the corpus to investigate on the distribution of PoS (Part of Speech) for the tokens annotated as events. Most events are expressed in verbs and nouns. Sum of the two PoS types covers about 93% of all the event tokens, which is split into about 65% and 28% for verb and nouns, respectively.

The experiment is designed to see the effect of the selection method by using the feature candidates generated by the work of Jeong & Myaeng (2013), which showed the best performance in TimeML event recognition and classification in the literature. It generates feature sets based on the same criteria of the proposed method using syntactic dependencies and WordNet hypernyms. To find the concept (i.e., synset) of a target word, we applied the word sense disambiguation module of BabelNet (Ponzetto & Navigli, 2010). We also used Stanford Parser (Klein & Manning,

2003) to get the syntactic dependency based features.

A maximum entropy (ME) classifier was used because it showed the best performance for the tasks at hand, according to the literature. We also considered SVM, another popular machine learning algorithm in natural language processing. The evaluation was done by 5-fold cross validation, and the data of each fold was randomly selected. For the classifier, we used the Mallet machine learning package (McCallum, 2002) and Weka (Witten, Frank, & Hall, 2011).

## 5.2 Evaluation

We first evaluated the proposed tree-based feature selection in comparison with two widely accepted feature selection methods: information gain (IG) and  $\chi^2$ . For each feature selection method, we chose the number of features that gave the best performance in F1. In Table 2, TSEL means the pure tree-based feature selection without the reselection process using  $\chi^2$  whereas TSEL+ $\chi^2$  means the proposed method followed by  $\chi^2$ .

Compared to  $\chi^2$ , TSEL dramatically reduced the feature space significantly by 73.93% and 54.42% for event recognition and type classification, respectively, but the decrease of effectiveness was insignificant for the both tasks. The decrease was compensated by the reselection process (hence the TSEL+ $\chi^2$  case) to the point of 1.26% improvement over the  $\chi^2$  case. For type classification, only 40.68% of the features required by  $\chi^2$  were enough to achieve the same level of effectiveness achieved  $\chi^2$ . Due to the decrease of feature space, the running times of classification tasks (except preprocessing) were also quite reduced. The time-savings by TSEL were about 40% and 45% of  $\chi^2$  in the recognition and the type classification.

Event Recognition (ME)				
	IG	$\chi^2$	TSEL	TSEL + $\chi^2$
# features	202,495	255,371	66,578 (-73.93%)	64,041 (-74.92%)
P	0.8878	0.8720	0.8664	0.8779
R	0.8413	0.8531	0.8571	0.8687
F1	0.8639	0.8624	0.8617 (-0.08%)	0.8733 (+1.26%)
T	4.12 s	4.14 s	2.52 s (-39.13%)	2.51 s (-39.37%)

<sup>3</sup> We use  $\chi^2$  for discussion instead of IG because it showed the better performance than IG for the verb and noun event classification, which is the main focus of the research.

Type Classification (ME)				
	IG	$\chi^2$	TSEL	TSEL + $\chi^2$
# features	291,408	267,226	121,793 (-54.42%)	108,705 (-59.32%)
P	0.8050	0.8117	0.7847	0.8411
R	0.6340	0.6199	0.6334	0.6026
F1	0.7094	0.7029	0.7010 (-0.28%)	0.7021 (-0.11%)
T	9.73 s	9.69 s	5.31 s (-45.20%)	5.28 s (-45.51%)

P: Precision, R: Recall  
T: Running Time of Classification  
(at PC with 3.0 GHz Core 2 Duo CPU and 8 GB memory)

Table 2. Comparisons in time and effectiveness for event recognition and type classification

Event Recognition (SVM)				
	IG	$\chi^2$	TSEL	TSEL + $\chi^2$
# features	202,495	255,371	66,578 (-73.93%)	64,041 (-74.92%)
P	0.8277	0.8048	0.7338	0.8128
R	0.8406	0.8592	0.8806	0.8576
F1	0.8341	0.8311	0.8005	0.8346

Type Classification (SVM)				
	IG	$\chi^2$	TSEL	TSEL + $\chi^2$
# features	291,408	267,226	121,793 (-54.42%)	108,705 (-59.32%)
P	0.6189	0.6179	0.6633	0.6833
R	0.6931	0.6531	0.6790	0.6700
F1	0.6539	0.6350	0.6711	0.6766

Table 3. Comparisons in effectiveness for event recognition and type classification using SVM classifier

Looking at the performance of different PoS types, we found that the performance of noun events was more meaningfully improved with a significantly reduced feature set. With the feature set reduction ratios of 81.66% and 81.50% for recognition and type classification, respectively, we achieved 6.85% and 3.94% of increase in F1<sup>4</sup>. For verbs, the numbers of features used for class recognition were also reduced significantly, but the F1 scores were slightly decreased. Our analysis shows that the increase in effectiveness for nouns is mainly attributed to the fact that the synsets of most nouns are located at a deep level of WordNet hierarchy. On the contrary, the hierarchy for verbs is not as deep as that of nouns. Note that the tree-based selection method is most helpful when heavy redundancy of features with a deep hierarchy causes a problem.

<sup>4</sup> The results are statistically significant with  $p < 0.05$ .

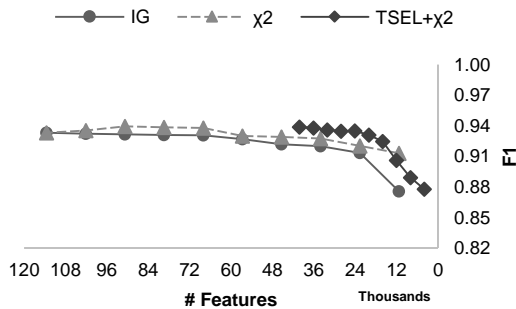
Recognition (ME)				
	Verb		Noun	
	# features	F1	# features	F1
$\chi^2$	90,792	0.9393	123,480	0.7138
TSEL	40,189 (-55.74%)	0.9385 (-0.09%)	25,169 (-79.62%)	0.7273 (+1.89%)
TSEL + $\chi^2$	40,180 (-55.74%)	0.9386 (-0.07%)	22,644 (-81.66%)	0.7627 (+6.85%)

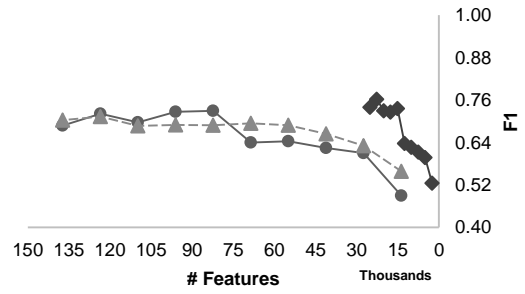
Classification (ME)				
	Verb		Noun	
	# features	F1	# features	F1
$\chi^2$	217,287	0.7406	47,080	0.6288
TSEL	99,100 (-54.39%)	0.7220 (-2.51%)	21,722 (-53.86%)	0.6149 (-2.21%)
TSEL + $\chi^2$	49,550 (-77.20%)	0.7223 (-2.47%)	8,708 (-81.50%)	0.6536 (+3.94%)

Table 4. Feature space sizes and effectiveness values for noun and verb events in event recognition and type classification

Figure 6 and 7 show the performance changes incurred by reducing the feature sets for different feature selection methods. The lines start from the point where all the selected features were used in each method and continue with a decrement of 10% of the feature set all the way to the minimum of 10% of the originally selected feature set. The starting points of TSEL+ $\chi^2$  indicate the results of pure TSEL. Despite the elimination of many features, the pure TSEL does not much harm the F1 compared to the best cases of IG and X2. It clearly shows that reducing the size of feature sets is less detrimental with the proposed method in almost all the cases than the other selection methods. TSEL also shows the possibility to select valuable features without manual check of performance for the feature space size. For event type classification, the manual selection process (TSEL+ $\chi^2$ ) is still needed in order to find the best features but it guarantees the more effectiveness.

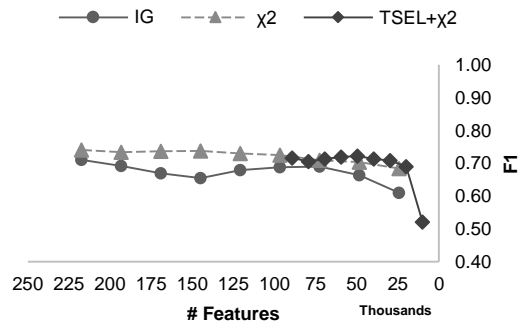


(a) Verb Event Recognition

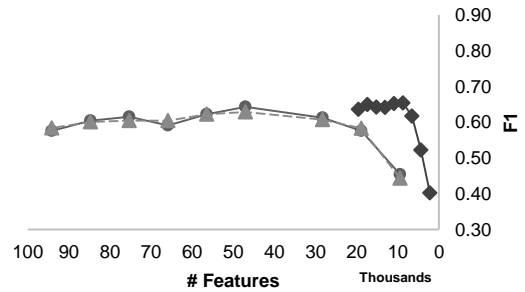


(b) Noun Event Recognition

Figure 6. Performance change with feature set reduction in event recognition in each of the feature selection methods



(a) Verb Event Type Classification



(b) Noun Event Type Classification

Figure 7. Performance change with feature set reduction in event type classification in each of the feature selection methods

For the type classification task, Table 5 shows detailed scores for all the event types separately. An improvement is observed for most of the event types except for OCCURRENCE. Our analysis shows that this is related to the size of the training data. Since the ratio of OCCURRENCE events is about 53% of all the events in the TimeBank corpus, the training data for the OCCURRENCE type is much bigger than the others. It indicates that the feature redundancy is problematic when the training data is relatively small and that careful selection of features is particularly important to avoid overfitting.

	$\chi^2$	TSEL+ $\chi^2$
REPORTING	0.9111	0.9201 (+0.99%)
PERCEPTION	0.6186	0.6292 (+1.71%)
ASPECTUAL	0.6444	0.6771 (+5.07%)
I_ACTION	0.6173	0.6346 (+2.80%)
I_STATE	0.6251	0.6866 (+9.84%)*
OCCURRENCE	0.7219	0.6980 (-3.31%)*
STATE	0.5246	0.5534 (+5.49%)*

Table 5. Performance for different event types (unit: F1). \* indicates that the percent increase or decrease is statistically significant with  $p < 0.05$ .

## 6 Related Work

EVITA (Saurí et al., 2005) is the first event recognition tool for TimeML specification. It recognizes events by using both linguistic and statistical techniques. It uses manually encoded rules based on linguistic information as main features to recognize events. It also uses WordNet classes to those rules for nominal event recognition, and checks whether the head word of noun phrase is included in the WordNet event classes. For sense disambiguation of nouns, it utilizes a Bayesian classifier trained on the SemCor corpus.

Boguraev & Ando (2007) analyzed the TimeBank corpus and presented a machine-learning based approach for automatic TimeML events annotation. They set out the task as a classification problem, and used a robust risk minimization (RRM) classifier to solve it. They used lexical and morphological attributes and syntactic chunk types in bi- and tri-gram windows as features.

Bethard & Martin (2006) developed a system, STEP, for TimeML event recognition and type classification. They adopted syntactic and semantic features, and formulated the event recognition task as classification in the word-chunking paradigm. They used a rich set of features: textual, morphological, syntactic dependency and some selected WordNet classes. They implemented a Support Vector Machine (SVM) model based on those features.

Llorens et al. (2010) presented an evaluation on event recognition and type classification. They added semantic roles to features, and built the Conditional Random Field (CRF) model to

recognize events. They conducted experiments about the contribution of semantic roles and CRF and reported that the CRF model improved the performance but the effects of semantic role features were not significant.

Jeong & Myaeng (2013) argued and demonstrated that unit feature dependency information and deep-level WordNet hypernyms are useful for event recognition and type classification. Their proposed method utilizes various features including lexical semantic and dependency-based combined features. In the TimeBank 1.2 corpus, the approach achieved 0.8601 and 0.7058 in F1 in event recognition and type classification, respectively.

## 7 Conclusion

In this paper, we proposed a novel feature selection method for event recognition and event type classification, which utilizes a semantic hierarchy of features. While our current work is based on the WordNet hierarchy and syntactic dependencies, the proposed method can be applied as long as it is possible to utilize a feature hierarchy, and shows the possibility to select valuable features without manual check of performance for the feature space size.

Our experimental results show that the proposed method is significantly effective in reducing the feature space compared to the well-known feature selection methods, and yet the overall effectiveness is similar to or sometimes better than a state-of-the-art approach depending on the PoS of the events. In particular, the effectiveness for noun events was improved quite meaningfully when the feature space was reduced significantly.

Although the proposed method showed the encouraging results, it still has some limitations. One issue is on the depth of the features in hierarchy. For verb, most features are located at shallow levels so the feature space reduction ratio is lower than those of noun. It implies that we need other approaches for verbs. Another one is on the recall. The proposed method showed high precision but relative lower recall. We conjecture that one reason is the lack of lexical information due to small size of TimeBank corpus.

Not only to improve recall but also for extensibility of the proposed method, we need to utilize other larger-scale resources for this tasks and even apply the proposed method for other types of text classification.

## Acknowledgments

This work was supported by a Microsoft Research Asia (MSRA) Faculty-Specific Project and by the research project of Korean Agency for Defense Development (ADD) [UD120064ED, Research on Extracting Contextual Factors and their Relations from Natural Language Factors].

## Reference

- Bethard, S., & Martin, J. H. (2006). Identification of event mentions and their semantic class. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 146–154). Association for Computational Linguistics.
- Boguraev, B., & Ando, R. (2007). Effective Use of TimeBank for TimeML Analysis. In F. Schilder, G. Katz, & J. Pustejovsky (Eds.), *Annotating, Extracting and Reasoning about Time and Events* (Vol. 4795, pp. 41–58). Springer Berlin Heidelberg.
- Daniel, N., Radev, D., & Allison, T. (2003). Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop* (Vol. 5, pp. 9–16). Association for Computational Linguistics.
- Jeong, Y., & Myaeng, S.-H. (2013). Using WordNet Hypernyms and Dependency Features for Phrasal-level Event Recognition and Type Classification. In P. Serdyukov, P. Braslavski, S. Kuznetsov, J. Kamps, S. Rüger, E. Agichtein, ... E. Yilmaz (Eds.), *Advances in Information Retrieval* (Vol. 7814, pp. 267–278). Springer Berlin Heidelberg.
- Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (pp. 423–430). Association for Computational Linguistics.
- Llorens, H., Saquete, E., & Navarro-Colorado, B. (2010). TimeML events recognition and classification: learning CRF models with semantic roles. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 725–733). Association for Computational Linguistics.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit.
- Ponzetto, S. P., & Navigli, R. (2010). Knowledge-rich Word Sense Disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 1522–1531). Association for Computational Linguistics.
- Pustejovsky, J. (2002). TERQAS: Time and Event Recognition for Question Answering Systems. In *Proceedings of ARDA Workshop*.
- Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., & Katz, G. (2003). TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of the 5th International Workshop on Computational Semantics*.
- Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., ... Lazo, M. (2003). The TIMEBANK Corpus. In *Proceedings of the Corpus Linguistics 2003 conference* (pp. 647–656).
- Pustejovsky, J., Knippen, R., Littman, J., & Saurí, R. (2007). Temporal and Event Information in Natural Language Text. In H. Bunt, R. Muskens, L. Matthewson, Y. Sharvit, & T. E. Zimmerman (Eds.), *Computing Meaning* (Vol. 83, pp. 301–346). Springer Netherlands.
- Saurí, R., Knippen, R., Verhagen, M., & Pustejovsky, J. (2005). Evita: a robust event recognizer for QA systems. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 700–707). Association for Computational Linguistics.
- Tufféry, S. (2011). *Data Mining and Statistics for Decision Making* (2nd ed.). John Wiley & Sons.
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann.

# Romanization-based Approach to Morphological Analysis in Korean SMS Text Processing

**Youngsam Kim**

Seoul National University/  
Gwanak-1, Gwanak-ro, Gwanak-gu,  
Seoul, South Korea  
youngsam@gmail.com

**Hyopil Shin**

Seoul National University/  
Gwanak-1, Gwanak-ro, Gwanak-gu,  
Seoul, South Korea  
hpshin@snu.ac.kr

## Abstract

In this research, we suggest an approach to retrieval-related tasks for Korean SMS text. Most of the previous approaches to such text used morphological analysis as the routine stage of the preprocessing workflow, functionally equivalent to POS tagging. However, such approaches suffer difficulties since Short Message Service language usually contains irregular orthography, atypically spelled words, unspaced segments, etc. Two experiments were conducted to measure how well these problems can be avoided with the transliteration of Korean to Roman letters. In summary, we will argue that such a Romanization-based retrieval method has several advantages since it provides an easier way to preprocess the data with a variety of linguistic rules.

## 1 Introduction

In this internet era, everyday people express opinions, comments, or sentiments; all of which can be accessed via the web. Particularly with the popularization of mobile computing devices, it has become easier than ever for people to share messages using social media services like Twitter or Facebook. However, such an environment brings new challenges for researchers who aim to analyze or interpret this linguistic data. One of the problems they encounter is that these written texts have a different form than those in published books or articles. They were often called as short message service language, txt-speak,

chat-speak, etc. This new data source has received attentions from various fields and researchers working in the field of sentiment analysis and opinion-mining often find that dealing with such texts using traditional approaches is problematic.

For agglutinative languages like Korean, since words are formed by combining lemmas and various affixes, morphological analysis is required to find the functional meaning of each component. Most previous studies used morphological analysis only to preprocess the text, but this approach exhibits several weaknesses when used on the data that is written in SMS-like languages. First of all, texts are often unspaced to save on typing time and sentence length (e.g., Twitter only allows 140 characters per tweet). Secondly, many words are not typed in the same way as their dictionary entries; the letters are changed or reduced to smaller units due to morpho-phonetic variation and abbreviation processes.

This paper will propose a new approach to overcome these shortcomings for morphologically rich languages while making use of Korean case studies. This approach adopts Yale Romanization to transliterate Korean alphabets into Roman letters, which, due to the way it handles Korean characters, allows for a more intuitive and easier way of implementing the relevant rewriting rules and handling morpho-phonetic changes.

In Section 2, the problems of morphological analysis will be described and the properties of Korean SMS language will be reviewed. This will be followed up in Section 3 by an introduction to the Romanization-based framework and the method of employing linguistic rules. Section 4 will detail two retrieval experiments which were prepared to show the effectiveness of this approach. The first experiment was designed to observe whether the Romanization method could

handle unspaced texts. The second experiment explored the possibility of covering phonetic variations of the target words using a small set of linguistic rules.

## 2 Related Research

Transliteration methods have often been used for the task of keyword matching across different languages (Chen and Ku, 2002; Fujii and Ishikawa, 2001). In contrast, Han (2006) applied the transliteration method to perform part-of-speech tagging for Korean texts using Xerox Finite State Tool. Similarly, this paper proposes using the method not for Korean-English word equivalents but for Korean-to-varied Korean word detection.

### 2.1 Problems of morphological analysis: lack of lexicon

As the number of the users using social networking services increases rapidly, sentiment analysis or opinion mining capable of automatically extracting the sentiment orientation from online posts has been gaining attention from NLP researchers (Hu and Liu, 2004; Kim and Hovy, 2004; Wiebe, 2000; Pak and Paroubek, 2010). As stated above, Korean is an agglutinative language and the chunks distinguished by space must be further separated into roots and affixes before they can be assigned a part-of-speech tag. This whole procedure is performed by morphological analysis and is critical to determining the meaning of a component. However, it is also known that such analysis can cause errors when not equipped with complete word entries to analyze the text. Such 'lack of lexicon' problems arise because after the morphological analysis categorizes all listed words in the sentence it classifies the remaining words as general nouns (Jang and Shin, 2010). Consider the following.

- (1) 너무 진부한 내용  
 nemu cinpuha-n nayyong  
 too stale-AD<sup>1</sup> content  
 'too stale contents'
- (2) 너무/a 진부/ncs 하/xpa ㄴ/exm 내용/nc  
 nemu/a<sup>2</sup>cinpu/ncs ha/xpa n/exm nayyong/nc

<sup>1</sup> Abbreviates: AD(adnominal suffix), NM(nominative particle), IN(instrumental particle), SC(subordinative conjunctive suffix), CP(conjunctive particle), PST(past tense suffix), DC(declarative final suffix), RE(retrospective suffix), CN(conjectural suffix), PR(pronoun), PP(propositive suffix), AC(auxiliary conjunctive suffix), GE (genitive particle), LC(Locative particle)

- (3) 너/npp 무진/nc 부/nc 한/nc 내용/nc  
 ne/npp mucin/nc pu/nc han/nc nayyong/nc  
 'you Mujin(place name) wealth resentment contents'

Sentence (3) is a misanalyzed version of sentence (1). The morphological analyzer's dictionary did not include the word entry ('cinbu') so the analyzer had to ignore the previous spacing and take the proper noun ('mucin') as a possible morpheme instead (Jang and Shin, 2010; p. 500).

As can be inferred from examples (1) ~ (3), typical morphological analysis consists of two stages: first, a sentence or clause is decomposed into relevant morphemes and then, second, the distinguished morphemes are assigned part-of-speech tags which denote grammatical function. The reason why the morpheme separation stage precedes POS tagging is to avoid the sparse data problem caused by the multiplicity of morphological variants of the same stem (Han and Palmer, 2005). However, the morpheme-based POS tagger in this process is vulnerable to irregular variations of word stems and, unfortunately, such variants are often found on the web. By the same reason it also produces erroneous results given unspaced texts since the complexity of the decomposing morphemes is very high.

This paper assumes that the morpheme analysis procedure is not feasible to process the SMS texts. In order to alleviate the pain, this research will focus on how one can extract the expected items from the linguistic data with which morpheme analysis does not work.

### 2.2 Properties of Korean SMS language

Socio-linguistic studies of the Korean SMS language have revealed that the irregular variations within the language are not arbitrarily irregular. The five distinguished properties have been summarized in Table 1 (Park, 2006; Lee, 2010; Kim, 2011).

Some of the properties in Table 1 can be found in English SMS texts as well, hinting that this set of the features may be due to common factors. 'Addition of sounds' is known as epenthesis phenomenon, existing in many languages including English; Crystal (2008) contended that many features of the texting language (logograms, initialisms, pictograms, abbreviations, nonstandard spellings) are not entirely new and have already been in writing systems for centuries.

<sup>2</sup> POS tags: a(adverb), ncs(stative common noun), xpa(adjective-derived suffix), exm(adnominal suffix), nc(common noun), npp(personal pronoun)

Properties	Examples
Ignoring spacing	그녀가 학교에 갔다. (spaced: ‘그녀가 학교에 갔다’) Ku nyeca-ka hakkyo-ey ka-ss-ta The woman(nyeca)-NM school(hakyo)-LC go-PST-DC ‘The woman went to school’
Linking sound or phonetic writing	멋있어 -> 머시써 mes-iss-e ‘gorgeous’ -> me-si-sse
Reductions or shortenings	메일 -> 멜 meyil ‘mail’ -> meyl 서울 -> 셀 sewul ‘Seoul’ -> sel
Acronyms or abbreviation	애니메이션 -> 애니 ay-ni-mey-i-syen ‘animation’ -> ay-ni 비밀번호 -> 비번 pi-mil-pen-ho ‘password’ -> pi-pen
Addition of sounds	아빠 -> 압빠 a-ppa ‘daddy’ -> ap-ppa 여보 -> 여봉 ye-po ‘honey’ -> ye-pong

Table 1. Summarization of properties in Korean SMS text

Ling and Baron (2007) reported that lexical shortening is the one of the most significant characteristics one can see in text messages. However, ‘ignoring spacing’ is the exception, since Korean suffixes can play as good predictors for the roles or the functions of the preceding stem. As such, removing spaces between phrases does not severely deteriorate the readers’ understanding given the content.

This study will focus on only three of the features presented in Table 1: Unspacing, Linking, and lexical reduction. According to linguistic analysis (Park, 2006; Lee, 2010), liaison and vowel reduction were very common among the phonetic variation of the words. Following that observation, this paper will incorporate a set of rules (presented in Park, 2006) in its experiment. Also, it will make use of the Romanization transliteration with the given phonological rules to cope with the lexical variations of the linguistic data.

### 3 Romanization-based morpheme retrieval process

This section will provide the detailed contents of the lexical variation generation process. Basically, the generation process consists of the three main sub-modules: word-ending addition, vowel-change rules, and vowel omission. Each of these modules contains a set of linguistic rules. As a result, each target word in the list obtains its variants. These variants can then be used to check the input sentence for derived forms of the target word.

#### 3.1 Yale Romanization

Yale Romanization is the transliteration systems developed at Yale University for Romanizing Mandarin, Cantonese, Korean, and Japanese. The Yale system of Korean<sup>3</sup> is generally used in linguistics and is adopted as the application of the transliteration process in this work. There are two other Romanization systems, Revised Romanization of Korean and McCune-Reischauer system, but since the emphasis of the systems is on how to transliterate entire Korean words to a string of elements of a pronounceable alphabet, only Yale Romanization has a one-to-one correspondence between Korean letters and English letters. Therefore, the other two systems are not considered in this study.

#### 3.2 Korean syllable

The Korean alphabet, called Hangeul, consists of blocks of multiple letters with each block representing a single syllable. For example, the first word of the Korean word, 한글 (hangeul), can be decomposed into three letters (‘ㅎ’/‘h’, ‘ㅏ’/‘a’, and ‘ㄴ’/‘n’) though it is represented as a single character (or block) in Korean orthography. One advantage of using Yale Romanization is the ability to linearize the Korean syllables into a sequence of the phonemes and thus allowing the linking of alphabets with their sound properties. The examples in Table 1 show this phenomenon

<sup>3</sup> <http://search.cpan.org/dist/Encode-Korean/lib/Encode/Korean/Yale.pm>



clearly. Although it seems ‘멧있어’(mes-iss-e) and ‘머시씨’(me-si-sse) have quite different word forms, their romanized forms are identical; implicating that the latter is the phonetic writing version of the former.<sup>4</sup> Morphological analysis has difficulty when analyzing such phonetically written words since it makes distinctions based on Hangul syllables instead of the string of the letters. That is, ‘mes-iss-e’ and ‘me-si-sse’ are discriminated because the hyphens are taken as the boundary of the syllables even though this is not the case during pronunciation.

### 3.3 Implementation of linguistic rules

#### 3.3.1 Conjugation of verbs and adjectives

In Korean grammar, verbs or adjectives do not come as independent morphemes, but always present along with an appropriate conjugation. This paper considers 17 word endings for the romanized target words, following the standard grammar of Korean (~다 ‘~ta’, ~은 ‘~un’, ~는 ‘~nun’, ~고 ‘~ko’, ~기 ‘~ki’, ~냐 ‘~nya’, ~었다 ‘~essta’, ~았다 ‘~assta’, ~든지 ‘~tunci’, ~던지 ‘~tenci’, ~지 ‘~ci’, ~게 ‘~key’, ~음 ‘~um’, ~ㅁ ‘~m’, ~습니 ‘~supni’, ~읍니 ‘~upni’, ~구 ‘~kwu’). When the target lexical entry is given with its part-of-speech information, and if it belongs to the categories of noun or adjective, the 17 endings are added to the base word, generating 17 different word forms to be included in the lexicon paradigm set.

#### 3.3.2 Vowel contraction or change

This paper accepted the five vowel variation rules from Park (2006) as follows:

- (4) ‘o’ + ‘a’ -> ‘wa’. e.g., pho-hang (‘Pho-hang’) -> phwang<sup>5</sup>
- (5) ‘wu’ + ‘e’ -> ‘ye’. e.g., swu-ep (‘a class’) -> syep
- (6) ‘wu’ + ‘i’ -> ‘wi’. e.g., pwi-in (‘wife’) -> pwin
- (7) ‘i’ + ‘a’ -> ‘ya’. e.g., ki-an (‘draft’) -> kyan

<sup>4</sup> It is worth to noting that it becomes easier to apply re-writing rules to the romanized Hangul text because of its’ linearity.

<sup>5</sup> Note that the rule of ‘H-weak’ is manipulated here and the rule functionally works by omitting any ‘h’ between of sonorants. This rule helps to capture the typical linking sound phenomenon in Korean.

- (8) ‘i’ + ‘e’ -> ‘ye’. e.g., ki-ek (‘memory’) -> kyek

The rules in (4) ~ (8) are supplied to the ‘vowel-change’ function that takes the Romanized target word as input and returns its changed form as the output.

#### 3.3.3 Vowel reduction

The vowel reduction rules used in this paper aim to catch two types of shortening; the first type is concerned with the middle syllable of the whole word while the second works on the last syllable. As described in section 3.2, one Hangul syllable consists of several letters and, if the syllable is the target area of the reduction process, the contained vowel may be removed. Therefore, considering the first word of the Korean word, 한글 (hangul), Romanized as ‘han’, if one omits the vowel (‘a’) then the result would be ‘hngul’.

Previous studies showed that Korean SMS language has frequent vowel reductions (Park, 2006; Lee, 2010; Kim, 2011) with the middle and final syllables being the most common targets for reduction. The example sentence (9) presents the omission of the vowel in the middle syllable and (10) provides an example of reduction in the final syllable.

- (9) sa-mwu-sil (‘office’) -> sam-sil
- (10) key-im (‘game’) -> keym

## 4 Experiment

Sentiment analysis or opinion mining techniques that utilize retrieval tasks to obtain the training sets or corpus data have to extract subjective chunks or morphemes from the real-world data. In fact, if one chooses to use an annotated subjective word list for the study, one must still go through the process of confirming whether the items in the given list are in the raw input data. For that reason, an effective retrieval operation is required for research which needs to manage unorganized message texts. This section documents two experiments. The first is on the effectiveness of the proposed approach for unspaced tweet texts, while the second focuses on lexical variation.

### 4.1 Data

A large tweet dataset was obtained from another study (Lee et al., 2011). This dataset contained 5,913,888 tweets from 11,379 users up

Method \ Condition	Spaced			Unspaced		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Romanization-based method	0.67	<b>0.79</b>	0.73	0.67	<b>0.79</b>	<b>0.73</b>
Morpheme analysis method	<b>0.94</b>	0.72	<b>0.82</b>	<b>0.95</b>	0.29	0.44

Table 2. Results of retrieval test for spacing factor

until the date of 14th Mar 2011. All the Twitter-specific components were filtered beforehand such as Twitter ID, Retweet marker, URL, and hash-tags. To form the list of the target sentiment words, 2823 sentiment word-morphemes, all annotated with their POS tags, were exploited from the previous study of the sentiment analysis on Korean movie reviews (Ko and Shin, 2010).

Since it is needed to construct the test dataset for the first experiment, 100 tweets were randomly selected from the tweet corpus and were manually annotated using the target sets found in the sentiment word list (as a result, 128 items were found in the 100 tweets).

For the second experiment, because no annotated corpus of Korean SMS texts was available, 80 tweets from the corpus were manually collected, each containing at least one irregular word (92 types in total). The varied word in the tweet was marked as the target and its corresponding original entry was restored and recorded in the target lexicon list.

#### 4.2 Experiment 1: Spaced vs. Unspaced

This experiment involved conducting a simple retrieval test for the selected 100 tweets using the sentiment word list as described above. To make a comparison with the proposed approach, the performance of the morphological analysis method also needed to be evaluated. As such, the data was tested using a Korean morphology analyzer.<sup>6</sup>

For the experimental conditions, one factor (spacing) was manipulated, providing two types of test dataset for the different approaches. Since removing all the spaces from the sentences would have left the morphological analyzer inoperable, only the spaces around the target were deleted to create the unspaced condition.

Table 2 shows the results of the retrieval experiment: how well each method found the target items and how many they picked incorrectly. The morpheme analysis-based approach barely chose any wrong targets, but it missed too many right

answers (the precision was 27% higher than the precision of Romanization-based method, while marking 7% lower recall rate). Although the morpheme analysis-based approach showed higher performance on the spaced text (0.82 versus 0.73 on F-Measure), the method proved ineffective against unspaced texts (the recall, compared to the Romanization method, was severely decreased from 0.72 to 0.29).

Following expectations, the Romanization-based method was very robust against unspaced texts. This phenomenon is easily explained by considering that the method searched for the target strings without any regard for morpheme boundaries. In contrast, the morpheme analysis-based method took the incoming chunks and separated them into morphemes, but when text is unspaced the morpheme analyzer has to perform word-segmentation as well as morpheme-analysis. Thus one would anticipate an increase in errors when the input text is not properly spaced, because it would increase the complexity of the analysis process.

However, unlike the predictions, the Romanization-based method recorded a lower precision than the morphological analysis-based approach. This result might be due to the set of short-length words in the target list. For example, words consisting of one or two letters such as ‘ak’ (both ‘evil’ or ‘music’ in English) may be erroneously identified in other words such as in ‘ak-ki’ (‘musical instrument’) since such short strings are likely to occur if only by chance. Thus, the Romanization-based method has a higher risk of errors if the system is supplied with such short terms. In the experiment above, the employed sentiment words were morphemes (not phrases or clauses), which is unfavorable for the Romanization approach. However, it is worthwhile to acknowledge that this is mitigated by employing the conjugation module, implying that well-defined rules can enhance performance.

<sup>6</sup> We used the Korean morpheme analyzer distributed from the 21st century Sejong Project ([http://www.sejong.or.kr/dist\\_frame.php](http://www.sejong.or.kr/dist_frame.php)).

Model	Precision	Recall	F-Measure
Vowel-reduction+, H weak+, Vowel-change+	0.80	<b>0.55</b>	<b>0.65</b>
Vowel-reduction+, H weak+, Vowel-change-	0.79	0.52	0.63
Vowel-reduction+, H weak-, Vowel-change+	0.79	0.53	0.63
Vowel-reduction-, H weak+, Vowel-change+	0.96	0.25	0.40
Vowel-reduction-, H weak-, Vowel-change+	0.96	0.23	0.37
Vowel-reduction-, H weak+, Vowel-change-	0.89	0.22	0.36
Vowel-reduction+, H weak-, Vowel-change-	0.78	0.5	0.61
Vowel-reduction-, H weak-, Vowel-change-	<b>1.0</b>	0.24	0.38
Morphological analysis-based method	<b>1.0</b>	0.067	0.13

Table 3. Results of retrieval tests for phonetically changed words

### 4.3 Experiment 2: Covering phonetic changes in the lexicon

Experiment 1 dealt with the cases where morpheme’s grammatical category information was given, allowing the use of conjugation rule functions. Experiment 2 considers the situation in which specific words or expressions are given without POS tags and with phonetic variations of the targets which must be resolved before its original can be retrieved from the tweet data.

A retrieval experiment was conducted given the test data as described in section 4.1. Unlike Experiment 1, this experiment utilized the sub-modules of the lexical shortening (as stated in section 3.3). The result is displayed in Table 3.

The numbers in bold of Table 3 refer to the highest values for the column (tied values are treated as the same). The conjugation function is not carried out here because of a lack of grammatical category information, thus only three kinds of functions were manipulated as above. While vowel-change rules only care about the replacement of vowels, vowel-reduction rules cope with the circumstances in which the vowels in the word are omitted, resulting in a shortened form. H-weak rule is the only component that relates to any consonant change phenomena in this system; removing the phoneme ‘h’ between word syllables under specific conditions (e.g., The Korean word, ‘coh-a’ meaning ‘good’ is reduced to ‘co-a’). The notation [+/-] indicates whether the mentioned function was employed in the construction of the target paradigm set.

As can be seen in Table 3, the full model (including all the three sub-modules) outperforms the other models, proving the research assumption that implementation of linguistic rules would cover a subset of the lexical variations in the SMS language. With capturing the case alone, even the weakest model (with neither vowel-reduction/change nor H-weak functions) showed better results than those of morphological analy-

sis. This is because it could find type-equivalence between tokens such as ‘cwuk-um’ (죽음, ‘death’) and ‘cwu-kum’ (죽음, ‘death’), obtaining the higher F-score (0.38 vs. 0.13).

Obviously, the strongest module affecting the results is the vowel-reduction function. Remember that this function has two omission rules for the middle and the last syllables of the target items.

The model (with vowel-reduction off and the other two functions on) clearly reveals the effect of this sub-module by exhibiting a rapid drop in F-score from 0.65 for the full-model to 0.40 for the current model.

This effect is due to the high frequency of the vowel-reduction variations. Table 4 summarizes the types of variation in the test data, providing an explanation for the results in Table 3. The proportion of phoneme reduction instances can be seen to be about a third of the total occurrences (36 out of 104, or approximately 35 percent), and it accounts for the steep decrease in F-score when the vowel-reduction function is not adopted. It is also worth noting that vowel-reduction in the first-syllable is quite rare; consistent with the linguistic analysis of empirical research (Park, 2006; p. 466). The creation of vowel-reduced forms clearly had a large effect, lowering the accuracy from 0.96 to 0.80. This is because the shortened targets can also be found as sub-string of bigger words. However, this shortcoming does not weaken the efficiency of the whole approach. The morphological analysis-based retrieval method found only a few items in the data, which was expected considering that this analysis is dependent on a syllable-based word lexicon.

In short, though a small set of the linguistic rules were employed, and even using them is still far from achieving complete coverage, the results of the experiment implicate that such a rule-based system can capture at least part of the vast, complicated range of linguistic variations.

Type	Specified type		Count
Linking Sound			8
Phoneme Reduction	Vowel reduction	Head-syllable vowel reduction	1
		Middle-syllable vowel reduction	17
		Final-syllable vowel reduction	14
		Others	4
	Consonant reduction	H-weak	9
		Others	5
Phoneme Change	Vowel change	22	
	Consonant change	11	
Abbreviation			5
Addition	Vowel addition	6	
	Consonant addition	2	
Total			104

Table 4. Types and counts of instances in test dataset of Exp. 2

## 5 Discussion and Conclusion

This paper confirmed that employing language-specific rules to handle SMS language text can enhance the results of the retrieval process. Although it is known that morphological analysis hardly produces erroneous results in formally written texts such as newspaper articles, the analysis results were made much worse for the SMS data in our experiments, which presented the motivation to pursue an additional approach. The procedure of sentiment analysis or opinion mining generally involves searching for items which are defined as subjectively meaningful, but typical morphological analysis cannot deal with the irregular changes of the web texts.

The reason why the morphological analysis does not work on such data is clear. The built-in stemmer or normalization process of the analyzer is not designed to cope with that kind of the text. However, in this paper, we tried to point out that judging the text as not well-formed enough to be processed is too quick. Instead, a set of generative rules to handle such texts were proposed and implemented in our experiments. Although those rules could be imported to a future morphological analyzer giving it broader coverage, suffice it to state that the text on the internet is not as simple as newspaper articles to the analyzers currently available.

For such a case, this proposed method could be an alternative way to preprocess Korean SMS texts and it should be noted that there could be similar approaches for other morphologically rich languages like Japanese or Turkish. Normalizing text is a very complicated task for the type of the languages and well-organized module would be needed if it has to manipulate SMS

texts for any morpheme-level retrieval process.

A Romanization transliteration scheme is used in this study because it naturally represents the phonetic properties of Korean syllables while providing a more intuitive way to apply a set of defined rules to the sequence. Since phonemic variation is quite common in SMS texts, as mentioned, this approach seems useful and practical regarding the results of the experiments. Although the size of the dataset which was used for the test is small, the sample set contained cases which were well known in previous literature and their linguistic patterns were consistent with reports (Park, 2006; Lee, 2010; Kim, 2011). However, to make the approach practical enough to be used by field engineers, a large scale corpus would be required to find the optimal set of the transformation rules, which is left for future study due to the lack of such annotated data at the time of writing.

### Acknowledgments

We would like to thank Lee, W., Cha, M. and Yang, H. for their kind approval to use the Tweet corpus and the three anonymous reviewers for their helpful comments.

### References

- Chen, H.-H., and Ku, L.-W. (2002). An NLP & IR approach to topic detection Topic detection and tracking (pp. 243-264): Kluwer Academic Publishers.
- Crystal, D. (2008). *Txtng: The Gr8 Db8*, Oxford University Press.
- Fujii, A., and Ishikawa, T. (2001). Japanese/English cross-language information retrieval: Exploration of query translation and transliteration. *Computers and the Humanities*, 35(4), 389-420.

- Han, N. R. (2006). Klex: A finite-state transducer lexicon of Korean. In *Finite-State Methods and Natural Language Processing* (pp. 67-77). Springer Berlin Heidelberg.
- Hu, M., and Liu, B. (2004). Mining and summarizing customer reviews. Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA.
- Jang, H., and Shin, H. (2010). Language-specific sentiment analysis in morphologically rich languages. Paper presented at the Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China.
- Kim, S. (2011). Phonological and Morphological Characters of Junmal in Korean Net Lingo. *Linguistics*. 61, 115-129. In Korean.
- Kim, S.-M., and Hovy, E. (2004). Determining the sentiment of opinions. Paper presented at the Proceedings of the 20th international conference on Computational Linguistics, Geneva, Switzerland.
- Ko, M., and Shin, H. (2010). Grading System of Movie Review through the Use of An Appraisal Dictionary and Computation of Semantic Segments. *Korean Journal of Cognitive Science*. 21(4), 669-696. In Korean.
- Lee, J. (2010). A Study of Phonological Features and Orthography in Computer Mediated Language. *Linguistic Research*, 27(1), 1-18. In Korean.
- Lee, W., Cha, M., Yang, H. (2011). Network Properties of Social Media Influentials : Focusing on the Korean Twitter Community. *Journal of Communication Research*. 48(2), 44-79. In Korean.
- Ling, R., and Baron, N.S. (2007). Text Messaging and IM: Linguistic Comparison of American College Data. *Journal of Language and Social Psychology*, 26(3), 291-298, doi:10.1177/0261927X06303480
- Pak, A., and Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Paper presented at the Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta. [http://www.lrec-conf.org/proceedings/lrec2010/pdf/385\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf)
- Park, C. (2006). A Phonological Study of PC Communication Language Noun. *Korean Education*, 119, 457-486. In Korean.
- Wiebe, J. M. (2000, July 30–August 3). Learning subjective adjectives from corpora. Paper presented at the In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, TX.

# Efficient Word Lattice Generation for Joint Word Segmentation and POS Tagging in Japanese

Nobuhiro Kaji\* and Masaru Kitsuregawa\*†

\*Institute of Industrial Science, The University of Tokyo

†National Institute of Informatics

{kaji, kitsure}@tkl.iis.u-tokyo.ac.jp

## Abstract

This paper investigates the importance of a word lattice generation algorithm in joint word segmentation and POS tagging. We conducted experiments on three Japanese data sets to demonstrate that the previously proposed pruning-based algorithm is in fact not efficient enough, and that the pipeline algorithm, which is introduced in this paper, achieves considerable speed-up without loss of accuracy. Moreover, the compactness of the lattice generated by the pipeline algorithm was investigated from both theoretical and empirical perspectives.

## 1 Introduction

Many approaches to joint word segmentation and POS tagging can be interpreted as reranking with a word lattice (Jiang et al., 2008), wherein a small lattice is generated for an input sentence, and then the lattice paths are reranked to obtain the optimal one. Examples of such a method include (Asahara and Matsumoto, 2000; Kudo et al., 2004; Kruengkrai et al., 2006; Jiang et al., 2008).

In such a framework, it is crucial to develop an efficient lattice generation algorithm. Since there are  $n_{+1}C_2 = O(n^2)$  word candidates, where  $n$  is the number of characters in the sentence, to be included in the lattice, it is prohibitively expensive to check all of them exhaustively. Such a naive method constitutes a severe bottleneck in a reranking system. Accordingly, in practice, it is necessary to resort to some technique to speed-up lattice generation.

It is, however, not straightforward to speed-up lattice generation for reranking, because there are

requirements that the lattice has to satisfy and it is necessary to achieve a speed-up while satisfying those requirements. Most importantly, the lattice should contain a sufficient amount of correct words; otherwise, the accuracy of the reranking system will be seriously degraded. Moreover, the lattice should be small: an excessively large lattice spoils the efficiency of the reranking system because it is expensive to find the optimal path of such a lattice.

For the reasons stated above, it is not readily obvious what sort of technique is effective for lattice generation. Despite its practical importance, this question, however, has not been well studied. For example, (Kudo et al., 2004) used a dictionary to filter word candidates. While indeed efficient, such a method is obviously prone to removing out-of-vocabulary (OOV) words from a lattice and degrade accuracy (Uchimoto et al., 2001). Jiang et al. (2008) employed a pruning-based algorithm to reduce the  $O(n^2)$  cost, but they did not investigate computational time required.

Given the above issues, the present study revisits lattice reranking by exploring the effectiveness of the lattice generation algorithm. Specifically, large-scale experiments were conducted on three Japanese data sets. The results of the experiments show that the pruning-based algorithm (Jiang et al., 2008) in fact incurs a non-negligible computational cost, which constitutes a bottleneck in the reranking system. Moreover, a pipelined lattice generation algorithm (see Section 3) was investigated as an alternative to the pruning-based one, and it was demonstrated that the reranking system using the pipeline algorithm speeds up the reranking more than 10 times without loss of accuracy. After that, the compactness of the lattice generated by the pipeline algorithm was examined from not

Input sentence: 東京都に住む (To live in Tokyo metropolis)

Word lattice:

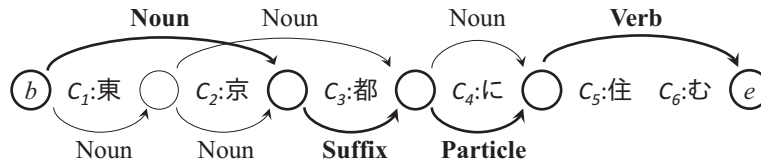


Figure 1: Example lattice (Kudo et al., 2004). The circle and arrow represent the node and edge, respectively. The bold edges represent the correct analysis.

only theoretical but also empirical perspectives.

The first contribution of this study is to shed light on the importance of the lattice generation algorithm in lattice reranking. As mentioned earlier, past studies paid little attention to elaborating the lattice generation algorithm. On the contrary, the results of our experiments reveal that the design of the lattice generation algorithm crucially affects the performance of the reranking system (including speed, accuracy, and lattice size).

The second contribution is to provide clear empirical evidence concerning the effectiveness of the pipeline algorithm. Although the pipeline algorithm itself is a simple application of well-known techniques (Xue, 2003; Peng et al., 2004; Neubig et al., 2011) and does not have much novelty, its effectiveness has been left unexplored in the context of lattice reranking. Consequently, its merits (or demerits) in relation to the pruning-based algorithm have also been unknown.

The third contribution is to develop an accurate reranking system based on the pipeline algorithm. The developed system achieved considerably higher F<sub>1</sub>-score than three software tools that are widely used in Japanese NLP (JUMAN<sup>1</sup>, MeCab<sup>2</sup>, and Kytea<sup>3</sup>), while achieving high speed close to two of the three.

## 2 Preliminaries

As a preliminary, a word lattice and lattice reranking for joint word segmentation and POS tagging are explained in Sections 2.1 and 2.2, respectively. After that, the pruning-based lattice generation algorithm proposed by Jiang et al. (2008) is introduced in Section 2.3.

<sup>1</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

<sup>2</sup><http://code.google.com/p/mecab>

<sup>3</sup><http://www.phontron.com/kytea>

### 2.1 Word lattice

A word lattice, or lattice for short, is a data representation that compactly encodes an exponentially large number of word segmentations and POS tagging results (Kudo et al., 2004; Jiang et al., 2008).

An example lattice is illustrated in Figure 1. A lattice is formally a directed acyclic graph. A node (a circle in Figure 1) corresponds to the position between two characters, representing a possible word boundary. Moreover, two special nodes,  $b$  and  $e$ , represent the beginning and ending of the sentence. An edge (an arrow) represents a word-POS pair  $(w, t)$ , where  $w$  is a word defined by two nodes, and  $t$  is a member of the predefined POS tag set.

Since every path from node  $b$  to  $e$  represents one candidate analysis of the sentence, the task of joint word segmentation and POS tagging can be seen as locating the most probable path amongst those in the lattice. Dynamic programming is usually used to locate the optimal path.

For later convenience, notations that will be used throughout this paper are introduced as follows.  $x$  and  $y$  are used to denote an input sentence and a lattice path. It is presumed that sentence  $x$  has  $n$  characters, and  $c_i$  is used to denote the  $i$ -th character ( $1 \leq i \leq n$ ).  $w$  and  $t$  are used to denote a word and a POS tag, respectively.

### 2.2 Lattice reranking

Lattice reranking is an approximate inference technique for joint word segmentation and POS tagging (Jiang et al., 2008). In this approach, a small lattice is generated for an input sentence, and the paths of the lattice are then reranked to obtain the optimal one. The advantage of this approach is that the search space is greatly reduced in the same manner as conventional list-based reranking (Collins, 2000), while an exponentially large num-

ber of candidates is maintained in the lattice (Jiang et al., 2008).

In this framework, the task of joint word segmentation and POS tagging can be formalized as

$$\hat{y} = \arg \max_{y \in L(x)} \text{SCORE}(x, y) \quad (1)$$

where  $\hat{y}$  is the optimal path,  $L(x)$  is the lattice created for sentence  $x$ , and  $\text{SCORE}(x, y)$  is a function for scoring path  $y$  of lattice  $L(x)$ . For notational convenience, lattice  $L(x)$  is treated as a set of paths.

In this paper we explore the algorithm for generating the lattice  $L(x)$ . A naive approach requires  $O(n^2)$  time to determine which word candidate to include in  $L(x)$ , as mentioned in Section 1, and constitutes a bottleneck. Although additional time is required to perform the  $\arg \max$  operation, it is practically negligible because the lattice generated in this framework is generally small.

### 2.3 Pruning-based algorithm

Jiang et al. (2008) proposed a pruning-based lattice generation algorithm for reranking. Here, we briefly describe their algorithm. Interested readers may refer to (Jiang et al., 2008) for its details.

The pruning-based algorithm generates a lattice, specifically the edge set  $E$  constituting a lattice, by considering each character in a left-to-right fashion (Algorithm 1). The algorithm enumerates word-POS pairs  $(w, t)$ , or edges, that end with the current character,  $c_i$ , and stores them in the candidate list,  $C$  (line 5-10). Top-scored  $k$  edges in  $C$  are then moved to  $E$  (line 11). Note that the word length  $l$  is limited to, at most,  $K$  characters (line 5).

This algorithm can be understood as pruning  $O(n^2)$  candidate space by setting threshold  $K$  on the maximum word length. Although this method is much more efficient than exhaustively searching over the entire candidates, it still incurs non-negligible computational overhead, as we will demonstrate in the experiments.

An additional issue involving the pruning-based algorithm is how to determine the value of  $K$ . Although a smaller value of  $K$  reduces computational cost more, it is prone to remove more correct word-POS pairs from the search space. While this trade-off was not investigated by Jiang et al. (2008), it is examined in our experiment (see Section 5).

---

#### Algorithm 1 Pruning-based lattice generation algorithm.

---

```

1:  $T \leftarrow$  a set of all POS tags
2:  $E \leftarrow \emptyset$ 
3: for  $i = 1 \dots n$  do
4:    $C \leftarrow \emptyset$ 
5:   for  $l = 1 \dots \min(i, K)$  do
6:      $w \leftarrow c_{i-l+1}c_{i-l+2} \dots c_i$ 
7:     for  $t \in T$  do
8:        $C \leftarrow C \cup (w, t)$ 
9:     end for
10:  end for
11:  add top- $k$  edges in  $C$  to  $E$ .
12: end for
13: return  $E$ 

```

---



---

#### Algorithm 2 Pipelined lattice generation algorithm.

---

```

1:  $E \leftarrow \emptyset$ 
2:  $W \leftarrow \text{WORDGENERATOR}(x)$ 
3: for  $w \in W$  do
4:    $T \leftarrow \text{POSTAGGENERATOR}(x, w)$ 
5:   for  $t \in T$  do
6:      $E \leftarrow E \cup (w, t)$ 
7:   end for
8: end for
9: return  $E$ 

```

---

## 3 Pipeline Algorithm

As an alternative to the pruning-based algorithm, a pipelined lattice generation algorithm, which generates words and POS tags independently, is proposed here. In a nutshell, this method first generates the word set  $W$  constituting the lattice (Algorithm 2 line 2), and it then generates POS tags for each of the words (line 4).

The advantage of this approach is that it can naturally avoid searching the  $O(n^2)$  candidate space by exploiting a character-based word segmentation model (Xue, 2003; Peng et al., 2004; Neubig et al., 2011) to obtain the word set  $W$ . This algorithm has linear-time complexity in the sentence length and hence is efficient.

This section proceeds as follows. Sections 3.1 and 3.2 describe how to generate words and POS tags, respectively. The computational complexity is then examined in Section 3.3.

### 3.1 Word generation

The character-based word segmentation model (Xue, 2003; Peng et al., 2004; Neubig et al., 2011) is used to generate word set  $W$  (Figure 2 line 2). This model performs segmentation by assigning tag sequence  $\mathbf{b}$  to the input sentence:

$$\mathbf{b} = \arg \max_{\mathbf{b}} \Lambda_w \cdot \mathbf{F}_w(x, \mathbf{b})$$



Name	Template
Char. $n$ -gram	$\langle c_{i-1}, b_i \rangle, \langle c_i, b_i \rangle, \langle c_{i+1}, b_i \rangle, \langle c_{i-2}, c_{i-1}, b_i \rangle, \langle c_{i-1}, c_i, b_i \rangle, \langle c_i, c_{i+1}, b_i \rangle, \langle c_{i+1}, c_{i+2}, b_i \rangle,$ $\langle c_{i-3}, c_{i-2}, c_{i-1}, b_i \rangle, \langle c_{i-2}, c_{i-1}, c_i, b_i \rangle, \langle c_{i-1}, c_i, c_{i+1}, b_i \rangle, \langle c_i, c_{i+1}, c_{i+2}, b_i \rangle, \langle c_{i+1}, c_{i+2}, c_{i+3}, b_i \rangle$
Char. type $n$ -gram	$\langle c'_{i-1}, b_i \rangle, \langle c'_i, b_i \rangle, \langle c'_{i+1}, b_i \rangle, \langle c'_{i-2}, c'_{i-1}, b_i \rangle, \langle c'_{i-1}, c'_i, b_i \rangle, \langle c'_i, c'_{i+1}, b_i \rangle, \langle c'_{i+1}, c'_{i+2}, b_i \rangle,$ $\langle c'_{i-3}, c'_{i-2}, c'_{i-1}, b_i \rangle, \langle c'_{i-2}, c'_{i-1}, c'_i, b_i \rangle, \langle c'_{i-1}, c'_i, c'_{i+1}, b_i \rangle, \langle c'_i, c'_{i+1}, c'_{i+2}, b_i \rangle, \langle c'_{i+1}, c'_{i+2}, c'_{i+3}, b_i \rangle$
Dictionary	$\langle \text{BEGIN}, b_i \rangle, \langle \text{END}, b_i \rangle, \langle \text{INSIDE}, b_i \rangle, \langle \text{BEGIN}, s, b_i \rangle, \langle \text{END}, s, b_i \rangle, \langle \text{INSIDE}, s, b_i \rangle$

Table 1: Feature templates of word generation.  $c_i$  and  $c'_i$  represent the target character and its type, respectively.  $c'_i$  specifically takes one of the following values: (1) Roman alphabet, (2) Chinese *kanji* characters, (3) Japanese *hiragana* characters, (4) Japanese *katakana* characters, (5) numerical symbols, or (6) others. The neighboring characters and their types are similarly referred to as  $c_{i-1}$ ,  $c_{i+1}$ ,  $c'_{i+1}$ , and so on.  $b_i$  is the tag ( $B$  or  $I$ ) given to the target character. BEGIN and END represent whether a word in a dictionary begins with or ends before the target character, respectively. INSIDE means that the target character is inside the word.  $s$  denotes the length (1, 2, 3, 4, or  $5 \leq$ ) of the word registered in the dictionary.

Name	Template
Word	$\langle w, t \rangle$
Word length	$\langle \text{LENGTH}(w), t \rangle$
Affix	$\langle c_i, t \rangle, \langle c_i, c_{i+1}, t \rangle, \langle c_{j-1}, t \rangle, \langle c_{j-2}, c_{j-1}, t \rangle$
Neighboring string	$\langle c_{i-1}, t \rangle, \langle c_{i-2}, c_{i-1}, t \rangle, \langle c_{i-3}, c_{i-2}, c_{i-1}, t \rangle, \langle c_j, t \rangle, \langle c_j, c_{j+1}, t \rangle, \langle c_j, c_{j+1}, c_{j+2}, t \rangle$
Dictionary	$\langle \text{DICT}(w, t) \rangle, \langle \text{DICT}(w, t), t \rangle$

Table 2: Feature templates of POS tag generation.  $w = c_i c_{i+1} \dots c_{j-1}$  represents the word string, and  $t$  represents the target POS tag.  $\text{LENGTH}(w)$  returns the length of the word  $w$  in the number of characters: 1, 2, 3, 4, or  $5 \leq$ .  $\text{DICT}(w, t)$  is an indicator representing that word  $w$  with POS tag  $t$  is registered in a dictionary. The features in the last row are fired only when the target word is found in a dictionary.

where  $\mathbf{b} = b_1 \dots b_n$  is the character-based tag sequence that encodes the segmentation results;  $b_i = B$  and  $b_i = I$  represent whether the  $i$ -th character is the beginning or inside of a word, respectively.  $\mathbf{\Lambda}_w$  and  $\mathbf{F}_w(x, \mathbf{b})$  are weight and feature vectors, respectively.

The model is trained with the averaged structured perceptron (Collins, 2002) due to its simplicity and efficiency. The features illustrated in Table 1, as well as tag bigrams, were used for the training. The features in Table 1 is basically taken from (Neubig et al., 2011). The first two rows represent character strings surrounding the target character; the last row represents dictionary-based features similar to those described in (Neubig et al., 2011). The dictionary-based features are fired if a string in a sentence is registered as a word in a dictionary, and they encode whether the string begins with or ends before the target character, or includes the target character.

$\alpha$ -best outputs of this segmentation model are used to obtain word set  $W$ :

$$W = \cup_{i=1 \dots \alpha} W_i$$

where  $W_i$  is a word set included in the  $i$ -th best output. Hyperparameter  $\alpha$  controls the size of

word set  $|W|$  and is tuned by using development data.

### 3.2 POS tag generation

To generate POS tags for each word (Figure 2 line 4), a linear model was used. Given sentence  $x$  and word  $w$ , it assigns the following score to each POS tag  $t$  (Neubig et al., 2011):

$$\mathbf{\Lambda}_t \cdot \mathbf{F}_t(x, w, t)$$

where  $\mathbf{\Lambda}_t$  and  $\mathbf{F}_t(x, w, t)$  are weight and feature vectors, respectively. Averaged perceptron was used for training (Freund and Schapire, 1999).

Table 2 shows the feature templates. Word string, word length, prefixes and suffixes up to length two were used, and the adjacent strings of the word up to length three were used. We also check the presence of the word in a dictionary.

For each word, top- $\beta$  tags were used as the POS tag set  $T$  (line 4). Hyperparameter  $\beta$  is also tuned by using development data.

### 3.3 Computational complexity

Unlike the pruning-based algorithm, the pipeline algorithm can generate words of arbitrary lengths. Nevertheless, it still only needs  $O(n)$  time. This

can be proved as follows. First, the word segmentation model takes  $O(n)$  time to output word set  $W$ , since this step can be efficiently performed by dynamic programming. In addition, since  $O(|W|) = O(n)$ , the outer loop of the algorithm requires  $O(n)$  time. This can be verified as

$$|W| = |\cup_{i=1 \dots \alpha} W_i| \leq \sum_{i=1 \dots \alpha} |W_i| \leq \alpha n$$

where  $|W_i| \leq n$ . Since the process in lines 4-7 is independent of  $n$ , the pipeline algorithm requires  $O(n)$  time.

It also follows from the above discussion that the lattice size, that is, the number of edges, is also linear in the sentence length, i.e.,  $O(|E|) = O(n)$ . Consequently, since the node degree is at most  $\alpha$  (i.e., not dependent on  $n$ ), the lattice path can be efficiently reranked in  $O(n)$  time by using dynamic programming.

## 4 Perceptron-based Reranker

This section presents our reranker. Since the main focus of this study is in not reranking but lattice generation, a perceptron-based reranker was developed by simply following the procedure proposed by (Huang, 2008).

The scoring function  $\text{SCORE}(x, y)$  in equation (1) is defined as follows:

$$\begin{aligned} \hat{y} &= \arg \max_{y \in L(x)} \text{SCORE}(x, y) \\ &= \arg \max_{y \in L(x)} \Lambda \cdot \mathbf{F}(x, y) \end{aligned}$$

where  $\Lambda$  is the weight vector and  $\mathbf{F}(x, y)$  is the feature vector.

### 4.1 Training

The averaged perceptron algorithm was used to train weight vector  $\Lambda$  (Huang, 2008). Note here two minor technical issues that have to be addressed before the perceptron algorithm can be used for training the reranker.

First, the generated lattice  $L(x)$  might not include the oracle path. This possibility is avoided by simply adding all the nodes and edges in the oracle lattice to  $L(x)$ . This approach worked reasonably well in our experiments, while having the advantage of being simpler than the alternative (Huang, 2008; Jiang et al., 2008).

Second, the same data should not be used for training the lattice generator (i.e., the two models

described in Sections 3.1 and 3.2) and reranker. If the same data were used, we will end up using injuriously better lattices when training the reranker than testing. To meet this requirement, the training data were split into ten subsets. During training of the reranker, the lattices of each subset were provided by the lattice generator trained by using the remaining nine subsets. During testing, on the other hand, the lattice generator trained by using the entire training data was used.

## 4.2 Features

The features used for training the reranker include those listed in Table 1 and Table 2, as well as POS tag bigrams. For the features in Table 1, BIES encoding (Nakagawa, 2004) is used. Since all those features can be factorized, the optimal path is located by using dynamic programming.

## 5 Experiment

The effectiveness of the lattice generation algorithm was investigated in the experiment described in the following. Sections 5.1, 5.2, and 5.3 explain our experimental setting: data sets, lattice generation algorithms to be compared, and hyperparameter tuning. The experimental results are reported in Section 5.4. The experiments were performed on a computer with 3.2 GHz Intel® Xeon™ CPU and 32 GB memory.

### 5.1 Data sets

Three evaluation data sets were developed from three corpora: Kyoto Corpus (KC) version 4.0 (Kurohashi and Nagao, 1998), Kyoto university NTT Blog Corpus (KNBC) version 1.0 (Hashimoto et al., 2011), and Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa, 2008). Each corpus was randomly split into three parts: training, development, and test set. The size of each data set is listed in Table 3.

JUMAN dictionary version 7.0<sup>4</sup> was used to extract the dictionary-based features in the experiments using KC and KNBC. Because BCCWJ adopts word segmentation criteria and a POS tag set different from those of the other two corpora, a different dictionary, UniDic version 1.3.12<sup>5</sup>, was used in the experiment using BCCWJ.

<sup>4</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?NLPresources>

<sup>5</sup><http://www.tokuteicorpus.jp/dist>

	KC					KNBC					BCCWJ				
	Time	#Cand.	F <sub>1</sub>	Size		Time	#Cand.	F <sub>1</sub>	Size		Time	#Cand.	F <sub>1</sub>	Size	
Pruning ( $K=5$ )	20	22	212	†97.25	356	1.2	1.3	137	†92.72	235	25	26	163	†97.33	276
Pruning ( $K=10$ )	31	32	400	97.92	88.9	2.0	2.1	250	93.48	235	43	44	301	†98.08	69.0
Pruning ( $K=20$ )	62	63	702	<b>97.94</b>	88.9	3.6	3.8	413	93.42	235	88	90	516	<b>98.18</b>	69.0
Pipeline	<b>1.8</b>	<b>2.6</b>	<b>30.4</b>	<b>97.94</b>	<b>60.8</b>	<b>0.12</b>	<b>0.18</b>	<b>24.8</b>	<b>93.92</b>	<b>99.2</b>	<b>2.3</b>	<b>3.1</b>	<b>23.3</b>	98.10	<b>46.6</b>

Table 4: Comparison of the reranking systems with the different lattice generation algorithms. Best-performing results in each metric are highlighted in bold font.

	Training	Development	Testing
KC	30,608	4028	3764
KNBC	3453	385	348
BCCWJ	47,547	6144	5741

Table 3: The number of sentences included in the three data sets.

## 5.2 Lattice generation algorithms

Two types of rerankers were implemented: one uses the pruning-based lattice generation algorithm, and the other uses the pipeline algorithm. All the rerankers were trained in the same manner as described in Section 4.

Although Jiang et al. (2008) fixed pruning threshold  $K$  as 20,  $K \in \{5, 10, 20\}$  was tested to examine the effect of this parameter. As a result, three rerankers that use the pruning-based algorithm were thus created.

The pruning-based algorithm uses a character-based model<sup>6</sup> to obtain top- $k$  edges (Figure 1 line 11). Although Jiang et al. (2008) proposed several features to train this model, they are simplistic compared with those used in the pipeline algorithm (i.e., Table 1 and 2). To make the comparison as fair as possible, the feature listed in Table 1 and BIES encoding were used (c.f., Section 4.2) were used. The features listed in Table 2 were not used, because they are not usable in a character-based model. It is considered that this feature set is comparable with that used by the pipeline algorithm, because the reranker using the pruning-based algorithm achieved comparable F<sub>1</sub>-score with the one using the pipeline algorithm when  $K$  is large (see Section 5.4).

## 5.3 Hyperparameter tuning

Hyperparameter  $k$  of the pruning-based algorithm was tuned with the development data. The tuning was done by searching over  $\{1, 2, 4, 8, 16, \dots, 256\}$  and selecting  $k$  that gen-

<sup>6</sup>Not detailed this model in this paper; refer to (Jiang et al., 2008) for details.

erated the lattice with the fewest edges amongst those covering at least  $\theta\%$  of the correct edges.

Since the pipeline algorithm also has hyperparameters  $(\alpha, \beta)$ , the hyperparameters were tuned in a similar manner by performing a grid search over  $\{1, 2, 4, 8, 16, \dots, 256\} \times \{1, 2, 4, 8, 16, \dots, 256\}$ .

The value of  $\theta$  was set as 99, 97, and 99 for the three data sets, respectively. A smaller value of  $\theta$  was used for KNBC because over 99% coverage could not be achieved in this data set.

## 5.4 Results

Table 4 summarizes the time in seconds spent on lattice generation, overall processing time spent on reranking, average number of candidates per sentence (see below), word-level F<sub>1</sub>-score in the joint task, and average lattice size per sentence, where lattice size refers to the number of edges in a lattice.

As for the pruning-based algorithm, the number of candidates refers to the number of words to be considered (Figure 1 line 6). As for the pipeline algorithm, it refers to the size of word set  $W$  (Figure 2). This number serves as an estimation of the computational cost. Notice that it corresponds to the time consumed by the two outer loops in Figure 1 or by the outer loop in Figure 2.

The symbol † is used to represent that the difference in F<sub>1</sub>-score from the best-performing system is statistically significant ( $p < 0.01$ ). Bootstrap resampling with 1,000 samples was used to test the statistical significance.

### 5.4.1 Runtime

Table 4 reveals that the reranking system using the pruning-based algorithm consumes the vast majority of the time for lattice generation. In other words, the pruning-based algorithm is not efficient enough. This inefficiency was not pointed out in previous studies, e.g., (Zhang and Clark, 2010; Sun, 2011).

The results in Table 4 also demonstrate that the reranker using the pipeline algorithm is an order of magnitude faster than the pruning-based algorithms. It is significantly faster than even the case that  $K = 5$ . This result indicates the importance of using an efficient lattice generation algorithm in the reranking system.

Table 4 also indicates that the number of the candidates roughly correlates with the actual computation time spent on lattice generation. This correlation confirms that the speed-up is achieved mainly by reducing the number of word candidates to be considered.

#### 5.4.2 F<sub>1</sub>-score

F<sub>1</sub>-score of the reranking systems was investigated next. The pipeline algorithm achieved comparable or higher F<sub>1</sub>-score than the pruning-based algorithm. This result shows that the speed-up does not come at the cost of accuracy.

It is crucial for the pruning-based algorithm to select an appropriate threshold value,  $K$ . If the value is too small, F<sub>1</sub>-score will significantly drop. In case that  $K = 5$ , F<sub>1</sub>-score was statistically significantly worse than that attained by the best-performing system for all three data sets ( $p < 0.01$ ). On the other hand, an excessively large value ( $K = 20$ ) does not contribute to the increase of F<sub>1</sub>-score so much, while it considerably degrades the speed.

#### 5.4.3 Lattice size

Table 4 shows that the pipeline algorithm usually generates smaller lattices than the pruning-based algorithm. This is because the pruning-based algorithm has no mechanisms to prune nodes (Jiang et al., 2008). To be more specific, the pruning-based algorithm always produces  $n + 1$  nodes for a sentence with  $n$  characters; hence, the lattice size is prone to grow large. The pipeline algorithm is, on the other hand, free from such a problem.

The coverage of the correct edges as the function of the average lattice size was investigated as follows (Figure 2). For the pruning-based algorithm, which has only one hyperparameter,  $k$ , the graph was drawn by changing  $k$  over  $\{1, 2, 4, 8, 16\}$ . Note that the graph for  $K = 10$  is omitted, because almost the same lattices are generated for  $K = 10$  and  $K = 20$ . For the pipeline algorithm,  $\alpha = 32$  is fixed and  $\beta$  is changed over  $\{1, 2, 4, 8, 16\}$  to draw the two-dimensional graphs. It is clear that the lattice generated by

	KC	KNBC	BCCWJ
JUMAN	†95.37	93.85	N/A
MeCab	†95.45	†91.60	†96.31
Kytea	†96.95	†90.91	†97.10
Our reranker	<b>97.94</b>	<b>93.92</b>	<b>98.10</b>

Table 5: Comparison of F<sub>1</sub>-score with that achieved by the existing software.

the pipeline algorithm generally achieves higher coverage, while having a smaller number of edges than the pruning-based algorithm.

As discussed in Section 3.3, the size of word set  $|W|$  is linear in the sentence length. This analysis empirically justified as follows. The number of words is illustrated in Figure 3 as a function of sentence length. The three graphs in the figure clearly illustrate that the number of words grows linearly with increasing sentence length.

## 6 Comparison with Existing Software

As an additional experiment, the proposed pipeline-algorithm-based reranking system was compared with three software tools popular in Japanese NLP: JUMAN, MeCab (Kudo et al., 2004), and Kytea (Neubig et al., 2011).

Table 5 compares the F<sub>1</sub>-score of the proposed system with that attained by the three tools. Bootstrap resampling with 1,000 samples was used for the statistical significance test. The symbol † indicates that the F<sub>1</sub>-score is significantly lower than that achieved by the proposed system ( $p < 0.01$ ). It is clear that the proposed system outperforms the existing tools in the case of two of the three data sets, while performing comparably with JUMAN in the case of KNBC. Note that JUMAN is a rule-based system and is not applicable to BCCWJ because of the discrepancy in the definition of the segmentation criteria and POS tag set.

The speeds of the algorithms were also investigated. The proposed system processed 1400 sentences in a second, while JUMAN, MeCab, and Kytea processed 2100, 29000, and 3200 sentences, respectively. This result demonstrates that the proposed reranking system using the pipeline algorithm successfully achieved speed close to the two of the three tools, while keeping considerably higher F<sub>1</sub>-score.

## 7 Related Work

Several methods, other than the pruning-based algorithm (Jiang et al., 2008), have been developed

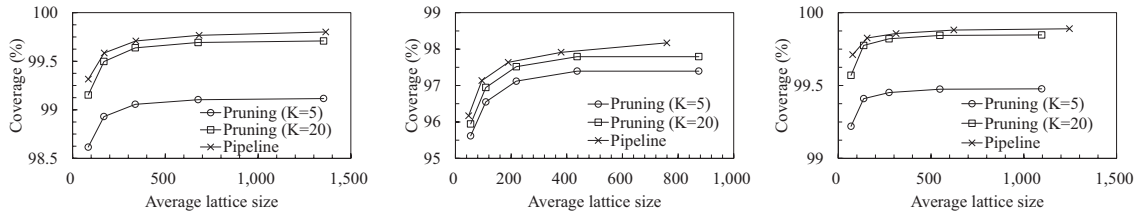


Figure 2: Coverage as the function of average lattice size (left: KC; middle: KNBC; right: BCCWJ).

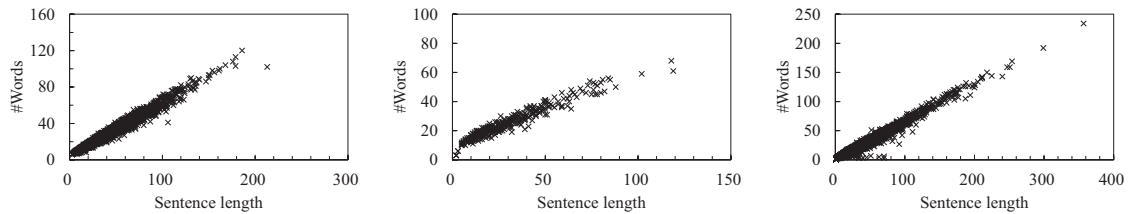


Figure 3: Number of words as a function of sentence length (left: KC; middle: KNBC; right: BCCWJ).

for lattice generation. However, they are dependent on an external dictionary and have limitations in handling OOV words. For example, Kudo et al. (2004) built a lattice based on dictionary-lookup. While efficient, such a method is prone to remove OOV words from a lattice and degrade accuracy (Uchimoto et al., 2001). Other researchers (Nakagawa and Uchimoto, 2007; Kruengkrai et al., 2009) used a word-character hybrid model, which combines dictionary-lookup and character-based modeling of OOV words. This method still has difficulty in using word-level information of OOV words.

The techniques utilized by the pipelined lattice generation algorithm have also been used elsewhere (Sassano, 2002; Peng et al., 2004; Shi and Wang, 2007; Neubig et al., 2011; Wang et al., 2011). However, the present study is the first to investigate the effectiveness of such a technique in the context of lattice reranking. Empirical studies similar to the ones made in this study are not found in the other work.

Zhang and Clark (2008) and Zhang and Clark (2010) proposed a fast decoding algorithm for joint word segmentation and POS tagging. The present study is largely complementary with theirs, since it did not investigate to improve decoding algorithm. Their algorithm should be useful for the decoding of our reranker especially when dynamic programming is not effective; for example, nonlocal features are used.

## 8 Conclusion

The effectiveness of the lattice generation algorithms used in joint word segmentation and POS tagging was investigated. While lattice generation has not been paid much attention to in previous studies, the present study demonstrated that the design of a lattice generation algorithm has a significant impact on the performance of a reranking system. It was showed that the simple pipeline algorithm outperforms the pruning-based algorithm. We hope that the pipeline algorithm serves as a simple but effective building block of future researches.

## Acknowledgments

This work was supported by the FIRST program. The authors thank the anonymous reviewers for their helpful comments.

## References

- Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of COLING*, pages 21–27.
- Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proceedings of ICML*, pages 175–182.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8.

- Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.
- Chikara Hashimoto, Sadao Kurohashi, Daisuke Kawahara, Keiji Shinzato, and Masaaki Nagata. 2011. Construction of a blog corpus with syntactic, anaphoric, and semantic annotations (in Japanese). *Journal of Natural Language Processing*, 18(2):175–201.
- Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL*, pages 586–594.
- Wenbin Jiang, Haitao Mi, and Qun Liu. 2008. Word lattice reranking for Chinese word segmentation and part-of-speech tagging. In *Proceedings of Coling*, pages 385–392.
- Canasai Kruengkrai, Virach Sorntlamvich, and Hitoshi Isahara. 2006. A conditional random field framework for Thai morphological analysis. In *Proceedings of LREC*, pages 2419–2424.
- Canasai Kruengkrai, Kiyooki Uchimoto, Jun’ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proceedings of ACL*, pages 513–521.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of EMNLP*, pages 230–237.
- Sadao Kurohashi and Makoto Nagao. 1998. Building a Japanese parsed corpus while improving the parsing system. In *Proceedings of LREC*, pages 719–724.
- Kikuo Maekawa. 2008. Balanced corpus of contemporary written Japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*, pages 101–102.
- Tetsuji Nakagawa and Kiyooki Uchimoto. 2007. A hybrid approach to word segmentation and POS tagging. In *Proceedings of ACL, Demo and Poster Sessions*, pages 217–220.
- Tetsuji Nakagawa. 2004. Chinese and Japanese word segmentation using word-level and character-level information. In *Proceedings of Coling*, pages 466–472.
- Graham Neubig, Yousuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust adaptable Japanese morphological analysis. In *Proceedings of ACL*, pages 529–533.
- Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of Coling*, pages 562–568.
- Manabu Sassano. 2002. An empirical study of active learning with support vector machines for Japanese word segmentation. In *Proceedings of ACL*, pages 505–512.
- Yanxin Shi and Mengqiu Wang. 2007. A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of IJCAI*, pages 1707–1712.
- Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL*, pages 1385–1394.
- Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. The unknown word problem: a morphological analysis of Japanese using maximum entropy aided by a dictionary. In *Proceedings of EMNLP*, pages 91–99.
- Yiou Wang, Jun’ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of IJCNLP*, pages 309–317.
- Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.
- Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL*, pages 888–896.
- Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS tagging using a single discriminative model. In *Proceedings of EMNLP*, pages 843–852.

# A Simple Approach to Unknown Word Processing in Japanese Morphological Analysis

Ryohei Sasano<sup>1</sup>

Sadao Kurohashi<sup>2</sup>

Manabu Okumura<sup>1</sup>

<sup>1</sup> Precision and Intelligence Laboratory, Tokyo Institute of Technology

<sup>2</sup> Graduate School of Informatics, Kyoto University

{sasano,oku}@pi.titech.ac.jp, kuro@i.kyoto-u.ac.jp

## Abstract

This paper presents a simple but effective approach to unknown word processing in Japanese morphological analysis, which handles 1) unknown words that are derived from words in a pre-defined lexicon and 2) unknown onomatopoeias. Our approach leverages derivation rules and onomatopoeia patterns, and correctly recognizes certain types of unknown words. Experiments revealed that our approach recognized about 4,500 unknown words in 100,000 Web sentences with only 80 harmful side effects and a 6% loss in speed.

## 1 Introduction

Morphological analysis is the first step in many natural language applications. Since words are not segmented by explicit delimiters in Japanese, Japanese morphological analysis consists of two subtasks: word segmentation and part-of-speech (POS) tagging. Japanese morphological analysis has successfully adopted lexicon-based approaches for newspaper articles (Kurohashi et al., 1994; Asahara and Matsumoto, 2000; Kudo et al., 2004), in which an input sentence is transformed into a lattice of candidate words using a pre-defined lexicon, and an optimal path in the lattice is then selected. Figure 1 shows an example of a word lattice for morphological analysis and an optimal path. Since the transformation from a sentence into a word lattice basically depends on the pre-defined lexicon, the existence of unknown words, i.e., words that are not included in the pre-defined lexicon, is a major problem in Japanese morphological analysis.

There are two major approaches to this problem: one is to augment the lexicon by acquiring unknown words from a corpus in advance (Mori and Nagao, 1996; Murawaki and Kurohashi, 2008) and the other is to introduce better unknown word processing to the morphological ana-

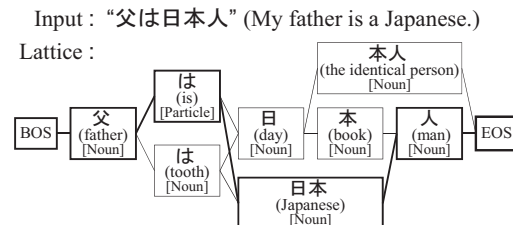


Figure 1: Example of word lattice. The bold lines indicate the optimal path.

lyzer (Nagata, 1999; Uchimoto et al., 2001; Asahara and Matsumoto, 2004; Azuma et al., 2006; Nakagawa and Uchimoto, 2007). Although both approaches have their own advantages and should be exploited cooperatively, this paper focuses only on the latter approach.

Most previous work on this approach has aimed at developing a single general-purpose unknown word model. However, there are several types of unknown words, some of which can be easily dealt with by introducing simple derivation rules and unknown word patterns. In addition, as we will discuss in Section 2.3, the importance of unknown word processing varies across unknown word types. In this paper, we aim to deal with unknown words that are considered important and can be dealt with using simple rules and patterns.

Table 1 lists several types of Japanese unknown words, some of which often appear in Web text. First, we broadly divide the unknown words into two classes: words derived from the words in the lexicon and the others. There are a lot of informal spelling variations in Web text that are derived from the words in the lexicon, such as “あなた” (y0u) instead of “あなた” (you) and “冷たーーい” (cooooool) instead of “冷たい” (cool). The types of derivation are limited, and thus most of them can be resolved by introducing derivation rules. Unknown words other than those derived from known words are generally difficult to resolve using only simple rules, and the lexicon augmentation approach would be better for them. However, this is not true for onomatopoeias. Although Japanese is rich in onomatopoeias and some of them do not



Unknown words derived from known words		
Type	Unknown word	Original word
<i>Rendaku</i> * (sequential voicing)	(たまご) ざけ ((tamago-)zake, sake-nog)	ざけ ( <i>sake</i> , Japanese alcoholic drink)
Substitution with long sound symbols*	ほんとう (troo)	ほんとう (true)
Substitution with lowercases*	あなた (y0u)	あなた (you)
Substitution with normal symbols	うれい (h@ppy)	うれしい (happy)
Insertion of long sound symbols*	冷た——い (cooooool)	冷たい (cool)
Insertion of lowercases*	冷たああい (cooooool)	冷たい (cool)
Insertion of vowel characters	冷たああい (cooooool)	冷たい (cool)

Unknown words other than those derived from known words		
Type	Unknown word	Corresponding English expression
Onomatopoeia with repetition*	かあかあ	caw-caw
Onomatopoeia w/o repetition*	シュツと	hiss
Rare word / New word	除染 / ツイッター	decontamination / Twitter

Table 1: Various types of Japanese unknown words. The “\*” denotes that this type is the target of this research. See Section 2.2 for more details.

appear in the lexicon, most of them follow several patterns such as ‘*ABAB*,’ ‘*AつBり*,’ and ‘*ABつと*,’<sup>1</sup> and they thus can be resolved by considering typical patterns.

Therefore, in this paper, we introduce derivation rules and onomatopoeia patterns to the unknown word processing in Japanese morphological analysis, and aim to resolve 1) unknown words derived from words in a pre-defined lexicon and 2) unknown onomatopoeias.

## 2 Background

### 2.1 Japanese morphological analysis

As mentioned earlier, lexicon-based approaches have been widely adopted for Japanese morphological analysis. In these approaches, we assume that a lexicon, which lists a pair consisting of a word and its corresponding part-of-speech, is available. The process of traditional Japanese morphological analysis is as follows:

1. Build a lattice of words that represents all the candidate sequences of words from an input sentence.
2. Find an optimal path through the lattice.

Figure 1 in Section 1 shows an example of a word lattice for the input sentence “父は日本人” (My father is Japanese), where a total of six candidate paths are encoded and the optimal path is marked with bold lines. The lattice is mainly built with the words in the lexicon. Some heuristics are also used for dealing with unknown words, but in most cases, only a few simple heuristics are used. In fact, the three major Japanese morphological analyzers, JUMAN (Kurohashi and Kawahara, 2005), ChaSen (Matsumoto et al., 2007),

<sup>1</sup>‘*A*’ and ‘*B*’ denote Japanese characters, respectively.

and MeCab (Kudo, 2006), use only a few simple heuristics based on the character types, such as hiragana, katakana, and alphabets<sup>2</sup>, that regard a character sequence consisting of the same character type as a word candidate.

The optimal path is searched for based on the sum of the costs for the path. There are two types of costs: the cost for a candidate word and the cost for a pair of adjacent parts-of-speech. The cost for a word reflects the probability of the occurrence of the word, and the connectivity cost of a pair of parts-of-speech reflects the probability of an adjacent occurrence of the pair. A greater cost means less probability. The costs are manually assigned in JUMAN, and assigned by adopting supervised machine learning techniques in ChaSen and MeCab, while the algorithm to find the optimal path is the same, which is based on the Viterbi algorithm.

### 2.2 Types of unknown words

In this section, we detail the target unknown word types of this research.

*Rendaku* (sequential voicing) is a phenomenon in Japanese morpho-phonology that voices the initial consonant of the non-initial portion of a compound word. In the following example, the initial consonant of the Japanese noun “ざけ” (*sake*, alcoholic drink) is voiced into “ざけ” (*zake*):

- (1) たまご ざけ (eggnog)  
*ta ma go - za ke.*

Since the expression “ざけ” (*zake*) is not included in a standard lexicon, it is regarded as an unknown word even if the original word “ざけ” (*sake*) is included in the lexicon. There are a lot

<sup>2</sup>Four different character types are used in Japanese: *hiragana*, *katakana*, Chinese characters, and Roman alphabet.



of studies on *rendaku* in the field of phonetics and linguistics, and several conditions that prevent *rendaku* are known, such as Lyman’s Law (Lyman, 1894), which stated that *rendaku* does not occur when the second element of the compound contains a voiced obstruent. However, few studies dealt with *rendaku* in morphological analysis. Since we have to check the adjacent word to recognize *rendaku*, it is difficult to deal with *rendaku* using only the lexicon augmentation approach.

Some characters are substituted by peculiar characters or symbols such as long sound symbols, lowercase *kana* characters<sup>3</sup>, in informal text. First, if there is little difference in pronunciation, Japanese vowel characters ‘あ’(a), ‘い’(i), ‘う’(u), ‘え’(e), and ‘お’(o) are sometimes substituted by long sound symbols ‘ー’ or ‘〜.’ For example, a vowel character ‘う’ in the Japanese adjective “ほんとう” (*hontou*, true) is sometimes substituted by ‘ー’ and this adjective is written as “ほんとー” (*hontô*, troo). We call this phenomenon **substitution with long sound symbols**. As well as long sound symbol substitution, some *hiragana* characters such as ‘あ’(a), ‘い’(i), ‘う’(u), ‘え’(e), ‘お’(o), ‘わ’(wa), and ‘か’(ka) are substituted by their lowercases: ‘あ,’ ‘い,’ ‘う,’ ‘え,’ ‘お,’ ‘わ,’ and ‘か.’ We call this phenomenon **substitution with lowercases**.

There are also other types of derivation, that is, some characters are inserted into a word that is included in the lexicon. In the following examples, long sound symbols and lowercase are inserted into the Japanese adjective “冷たい” (cool).

- (2) 冷たーーーい (Insertion of  
(coooooo) long sound symbols)
- (3) 冷たあああい (Insertion of lowercases)  
(coooooo)

In addition to the unknown words derived from words in the lexicon, there are several types of unknown words that contain rare words such as “除染” (decontamination), new words such as “ツイッター” (Twitter), and onomatopoeias such as “かああ” (caw-caw). We can easily generate Japanese onomatopoeias that are not included in the lexicon. Most of them follow several patterns, such as ‘ABAB,’ ‘AㄗBり,’ and ‘ABつと,’ and we classified them into two types, **onomatopoeias with repetition** such as ‘ABAB,’ and **onomatopoeias without repetition** such as ‘AㄗBり.’

<sup>3</sup>In this paper, we call the following characters lowercase: ‘あ,’ ‘い,’ ‘う,’ ‘え,’ ‘お,’ ‘わ,’ and ‘か.’

### 2.3 Importance of unknown word processing of each type

The importance of unknown word processing varies across unknown word types.

We give three example sentences (4), (5), and (6), which include the unknown words “もこもこ” (fluffy), “除染” (decontamination), and “ツイッター” (Twitter), respectively. In these examples, (a) denotes the desirable morphological analysis and (b) is the output of our baseline morphological analyzer, JUMAN version 5.1 (Kurohashi and Kawahara, 2005).

- (4) Input: ふわふわでもこもこの肌触り。  
(A soft and fluffy feeling to the touch.)  
(a) ふわふわ / で / もこもこ / の / 肌触り。  
soft and fluffy of touch  
(b) ふわふわ / でも / こもこ / この肌触り。  
soft but straw matting this touch
- (5) Input: 除染が必要。  
(Decontamination is required.)  
(a) 除染 / が / 必要。  
decontamination is required  
(b) 除 / 染 / が / 必要。  
UNKNOWN WORD UNKNOWN WORD is required
- (6) Input: 昨日、ツイッターを始めた。  
(I started Twitter yesterday.)  
(a) 昨日、 / ツイッター / を / 始めた。  
yesterday Twitter ACC started  
(b) 昨日、 / ツイッター / を / 始めた。  
yesterday UNKNOWN WORD ACC started

In the case of (4), the unknown word “もこもこ” (fluffy) is divided into three parts by JUMAN, and influences the analyses of the adjacent function words, that is, “で” (and) is changed to “でも” (but) and “の” (of) is changed to “この” (this), which will strongly affect the other NLP applications. The wide scope of influence is due to the fact that “もこもこ” consists of *hiragana* characters like most Japanese function words. On the other hand, in the case of (5), although the unknown word “除染” (decontamination) is divided into two parts by JUMAN, there is no influence on the adjacent analyses. Moreover, in case of (6), although there is no lexical entry of “ツイッター” (Twitter), the segmentation is correct thanks to simple character-based heuristics for out-of-vocabulary (OOV) words.

These two unknown words do not contain *hiragana* characters, and thus, we think it is important to resolve unknown words that contain *hiragana*. Since unknown words derived from words in the lexicon and onomatopoeias often contain *hi-*

*ragana* characters, we came to the conclusion that it is more important to resolve them than to resolve rare words and new words that often consist of *katakana* and Chinese characters.

## 2.4 Related work

Much work has been done on Japanese unknown word processing. Several approaches aimed to acquire unknown words from a corpus in advance (Mori and Nagao, 1996; Murawaki and Kurohashi, 2008) and others aimed to introduce better unknown word model to morphological analyzer (Nagata, 1999; Uchimoto et al., 2001; Asahara and Matsumoto, 2004; Nakagawa and Uchimoto, 2007). However, there are few works that focus on certain types of unknown words.

Kazama et al. (1999)’s work is one of them. Kazama et al. improved the morphological analyzer JUMAN to deal with the informal expressions in online chat conversations. They focused on substitution and insertion, which are also the target of this paper. However, while our approach aims to develop heuristics to flexibly search the lexicon, they expanded the lexicon, and thus their approach cannot deal with an infinite number of derivations, such as “冷たい—い,” and “冷—たい—” for the original word “冷たい.” In addition, Ikeda et al. (2009) conducted experiments using Kazama et al.’s approach on 2,000,000 blogs, and reported that their approach made 37.2% of the sentences affected by their method worse. Therefore, we conjecture that their approach only benefits a text that is very similar to the text in online chat conversations.

Kacmarcik et al. (2000) exploited the normalization rules in advance of morphological analysis, and Ikeda et al. (2009) replaced peculiar expressions with formal expressions after morphological analysis. In this research, we exploit the derivation rules and onomatopoeia patterns in morphological analysis. Owing to such a design, our system can successfully deal with *rendaku*, which has not been dealt with in the previous works.

UniDic dictionary (Den et al., 2008) handles orthographic and phonological variations including *rendaku* and informal ones. However, the number of possible variations is not restricted to a fixed number because we can insert any number of long sound symbols or lowercases into a word, and thus, all the variations cannot be covered by a dictionary. In addition, as mentioned above, since we

Input: “おいしかったで—す” (おいしかったです, It was delicious)

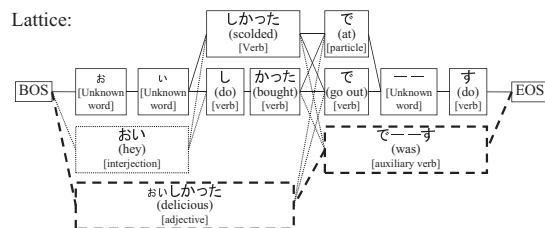


Figure 2: Example of a word lattice with new nodes “おい,” “おいしかった,” and “で—す.” The broken lines indicate the added nodes and paths, and the bold lines indicate the optimal path.

have to take into account the adjacent word to accurately recognize *rendaku*, the lexical knowledge alone is not sufficient for *rendaku* recognition.

For languages other than Japanese, there is much work on text normalization that aims to handle informal expressions in social media (Beaufort et al., 2010; Liu et al., 2012; Han et al., 2012). However, their target languages are segmented languages such as English and French, and thus they can focus only on normalization. On the other hand, since Japanese is an unsegmented language, we have to also consider the word segmentation task.

## 3 Proposed Method

### 3.1 Overview

We use the rule-based Japanese morphological analyzer JUMAN version 5.1 as our baseline system. Basically we only improve the method for building a word lattice and do not change the process for finding an optimal path from the lattice. That is, our proposed system only adds new nodes to the word lattice built by the baseline system by exploiting the derivation rules and onomatopoeia patterns. If the new nodes and their costs are plausible, the conventional process for finding the optimal path will select the path with added nodes.

For example, if a sentence “おいしかったで—す.” is input into the baseline system, it builds the word lattice that is described with solid lines in Figure 2. However, this lattice does not include such expressions as “おいしかった” and “で—す” since they are not included in the lexicon. Our proposed system transforms the informal expressions into their standard expressions such as “おいしかった” (delicious) and “です” (was) by exploiting the derivation rules, adds their nodes into the word lattice, and selects the path with these added nodes.

### 3.2 Resolution of unknown words derived from words in the lexicon

We deal with five types of unknown words that are derived from words in the lexicon: *rendaku*, substitution with long sound symbols, substitution with lowercases, insertion of long sound symbols, and insertion of lowercases. Here, we describe how to add new nodes into the word lattice.

**Rendaku** The procedure to add unvoiced nodes to deal with *rendaku* differs from the others. Since only the initial consonant of a word is voiced by *rendaku*, there is at most one possible voiced entry for each word in the lexicon. Hence, we add the voiced entries into the trie-based lexicon in advance if the original word does not satisfy any conditions that prevent *rendaku* such as Lyman’s Law.

For example, our system creates the entry “ざけ” (zake) from the original word “さけ” (sake), and adds it into the lexicon. When the system retrieves words that start from the fourth character in the example (1) in Section 2.2, “たまござけ,” the added entry “ざけ” (zake) is retrieved. Since *rendaku* occurs for the initial consonant of the non-initial portion of a compound word, our system adds the retrieved word only when it is the non-initial portion of a compound word.

#### Substitution with long sound symbols and lowercases

In order to cope with substitution with long sound symbols and lowercases, our system transforms the input text into normalized strings by using simple rules. These rules substitute a long sound symbol with one of the vowel characters: ‘あ,’ ‘い,’ ‘う,’ ‘え,’ and ‘お,’ that minimizes the difference in pronunciation. These rules also substitute lowercase characters with the corresponding uppercase characters. For example, if the sentence “ほんとうにおいしい。” (It is trooly DELicious.) is input, the nodes generated from the normalized string “ほんとうにおいしい。” are added to the word lattice along with the nodes generated from the original string.

#### Insertion of long sound symbols and lowercases

In order to cope with the insertion of long sound symbols and lowercases, our system transforms the input text into a normalized string using simple rules. These rules delete long sound symbols and lowercase characters that are considered to be inserted to prolong the original word pronunciation. For example, if the sentence “冷たあぁーいであーす。” (It iiiiss coooool.) is input, the nodes generated from the normalized string “冷

Pattern	Example	Transliteration
<i>ABAB</i>	たゆたゆ	tayu-tayu
<i>ABCABC</i>	ぽっかぽっか	pokka-pokka
<i>ABCDABCD</i>	ちよろりちよろり	chorori-chorori

Table 2: Onomatopoeia patterns with repetition and their examples. ‘A,’ ‘B,’ ‘C,’ and ‘D’ denote either *hiragana* or *katakana*. We consider only repetitions of two to four characters.

Pattern	Example	Transliteration
$H_1\text{っ}H_2$ り	ぽっこり	pokkori
$K_1\text{ッ}K_2$ リ	マッターリ	mattari
$H_1\text{っ}H_2Y$ り	ぺっちやり	pecchari
$K_1\text{ッ}K_2Y$ リ	ポッチャリ	pocchari
$K_1K_2\text{っ}と$	チラっと	chiratto
$K_1K_2\text{ッ}と$	パキッと	pakitto

Table 3: Onomatopoeia patterns without repetition and their examples. ‘H,’ denotes the *hiragana*, ‘K’ denotes the *katakana*, and ‘Y’ denotes the palatalized consonants such as ‘ゃ.’

たいです。” are added into the word lattice. We do not consider partly deleted strings such as “冷たあいでーす。” and the combination of substitution and insertion to avoid combinatorial explosion. Therefore, our system cannot deal with unknown words generated by both insertion and substitution, but such words are rare in practice.

**Costs for additional nodes** Our system imposes small additional costs to the node generated from the normalized string to give priority to the nodes generated from the original string. We set these costs by using a small development data set.

### 3.3 Resolution of unknown onomatopoeias

There are many onomatopoeias in Japanese. In particular, there are a lot of unfamiliar onomatopoeias in Web text. Most onomatopoeias follow limited patterns, and we thus can easily produce new onomatopoeias that follow these patterns. Hence, it seems more reasonable to recognize unknown onomatopoeias by exploiting the onomatopoeia patterns than by manually adding lexical entries for them.

Therefore, our system lists onomatopoeia candidates by using onomatopoeia patterns, as shown in Tables 2 and 3, and adds them into the word lattice. Figure 3 shows examples. The number of potential entries of onomatopoeias with repetition is large, but the candidates of onomatopoeias with repetition can be quickly searched for by using a simple string matching strategy. On the other hand, to search the candidates of onomatopoeias without repetition is a bit time consuming com-

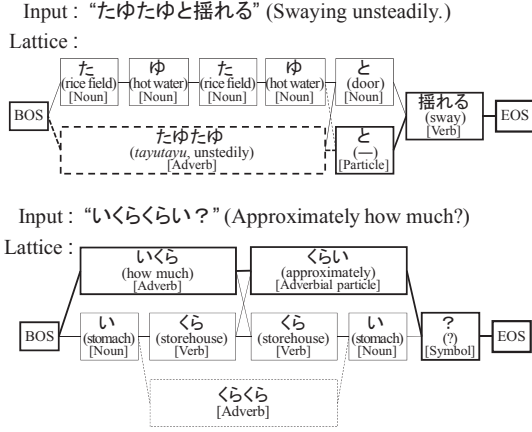


Figure 3: Examples of a word lattice with new nodes of onomatopoeia. The broken lines indicate the added nodes and paths, and the bold lines indicate the optimal path. While the optimal path includes the added node in the upper example, it does not in the lower example.

pared with trie search. However, the number of potential entries of onomatopoeias without repetition is not so large, and thus our system adds all possible entries of onomatopoeias without repetition into the trie-based lexicon in advance.

## 4 Experiments

### 4.1 Setting

We used 100,000 Japanese sentences to evaluate our approach. These sentences were obtained from an open search engine infrastructure TSUBAKI (Shinzato et al., 2008), which included at least one *hiragana* character and consisted of more than twenty characters

We first estimated the recall. Since it is too costly to create a set of data with all unknown words annotated, we made a set of data with only our target unknown words annotated. We could apply a set of regular expressions to reduce the unknown word candidates by limiting the type of unknown words. We manually annotated 100 expressions for each type, and estimated the recall.

A high recall, however, does not always imply that the proposed system performs well. It might be possible that our proposed method gives bad effects on non-target words. Therefore, we also compared the whole analysis with and without the rules/patterns from the following seven aspects:<sup>4</sup>

<sup>4</sup>There are two major reasons why we did not use the precision, recall and F-measure metrics to evaluate the overall performance. The first reason is that to create a large set of annotated data is too costly. The second reason, which is more essential, is that there is no clear definition of Japanese

1. The number of positive changes for 100 different outputs:  $P_{100D}$ .
2. The number of negative changes for 100 different outputs:  $N_{100D}$ .
3. The number of different outputs for 100,000 sentences:  $D_{100kS}$ .
4. The estimated number of positive changes for 100,000 sentences:  $P_{100kS}^*$ .
5. The estimated number of negative changes for 100,000 sentences:  $N_{100kS}^*$ .
6. The relative increase of the nodes:  $Node_{inc}$ .
7. The relative loss in speed:  $SP_{loss}$ .

Different outputs indicate cases in which the systems with and without rules/patterns output a different result. First, for each type of rule/pattern, we extracted 100 different outputs and manually classified them into three categories: the system with the rules/patterns was better (positive), the system without the rules/patterns was better (negative), and both outputs were undesirable (others). When these outputs differed in word segmentation, we only compared the segmentation but did not take into account the POS tags. On the other side, when these outputs did not differ in word segmentation, we compared the POS tags. Tables 6-10 list several examples. For example, “面白がれる” (can feel amused) in Table 6 should be analyzed as one word, but both systems with and without rules for *rendaku* divided it into several parts, and such a case is labeled as others.

We counted the number of different outputs for 100,000 sentences. We then calculated the estimated numbers of positive/negative changes for the sentences by using the equations:

$$X_{100kS}^* = D_{100kS} \times X_{100D}/100.$$

We also counted the number of created nodes in lattice and calculated the relative increase, which would affect the time for finding the optimal path from the word lattice, and measured the analysis time and calculated the relative loss in speed.

### 4.2 Results and Discussion

Table 4 lists the recall of our system for each unknown word type with the number of words that are covered by the UniDic dictionary. Note that while our system’s recall denotes the ratio of actually recognized words, the coverage of UniDic word segmentation, especially for unknown words. That is, we can accept various word boundaries. We thought it is more straight-forward and efficient to compare the differences between a baseline system and the proposed system.



Unknown word type	Recall of our system	# of words in UniDic
<i>Rendaku</i> (sequential voicing)	83/100	95
Substitution with long sound symbols	99/100	67
Substitution with lowercases	100/100	84
Insertion of long sound symbols	96/100	50
Insertion of lowercases	96/100	73
Onomatopoeia with repetition	89/100	78
Onomatopoeia w/o repetition	94/100	47

Table 4: Recall of our system and the coverage of UniDic.

only denotes the number of words included in the dictionary, which can be interpreted as the upper bound of the system based on UniDic. We can confirm our system achieved high recall for each type of unknown word. Since UniDic covered 95% of unknown words of *rendaku* type, we would be able to improve the *rendaku* recognition by incorporating UniDic and our approach that takes into account the adjacent word. Except for *rendaku*, our system’s recall was higher than the coverage of UniDic, which confirms the effectiveness of our method.

Table 5 summarizes the comparison between the analyses with and without the rules/patterns. In short, our method successfully recognized all types of unknown words with few bad effects. By introducing all the derivation rules and onomatopoeia patterns, there are 4,560 improvements for 100,000 sentences with only 80 deteriorations and a 6.2% loss in speed. In particular, the derivation rules of insertion and substitution of long sound symbols and lowercases produced 3,327 improvements for 100,000 sentences at high recall values (see Table 4) with only 27 deteriorations and a 3.8% loss in speed. We confirmed from these results that our approaches are very effective for unknown words in informal text. Since the number of newly added nodes was small, the speed loss is considered to be derived not from the optimal path searching phase but from the lattice building phase.

Table 6 lists some examples of the changed outputs by introducing the derivation rules for *rendaku*. As listed in Table 4 and 5, the *rendaku* processing produced more negative changes and the lower recall value compared with the other types. This indicates that *rendaku* processing is more difficult than resolving informal expressions with long sound symbols or lowercases. Since long sound symbols and lowercases rarely appear in the lexicon, there are few likely candidates other than the correct analysis. On the other hand, voiced characters often appear in the lexicon and formal

Our system	Baseline	Gold standard
Positive		
Input: 洗濯ばさみ (clothespin)		
洗濯 <b>ば</b> さみ	洗濯/ば/さみ	洗濯 <b>ば</b> さみ
Negative		
Input: 借入れがない方 (the man without)		
借入れ <b>が</b> ない	借 <b>入</b> れ <b>が</b> ない	借 <b>入</b> れ <b>が</b> ない
Others		
Input: 面白 <b>か</b> れる (can feel amused)		
面白 <b>か</b> れる	面白/か/れる	面白 <b>か</b> れる

Table 6: Examples of different outputs by introducing the derivation rule for *rendaku*. The ‘/’ denotes the boundary between words in the corresponding analysis, and the bold font indicates the correct output, that is, the output is the same as the gold standard.

Our approach	Baseline	Gold standard
Positive (insertion)		
Input: 苦 <b>い</b> 経験 (a bitter experiment)		
苦 <b>い</b> 経験	苦/い/経験	苦 <b>い</b> 経験
Positive (substitution)		
Input: おめでと <b>と</b> (congratulations)		
おめでと <b>と</b>	おめで/と/	おめでと <b>と</b>
Negative (substitution)		
Input: OK だ <b>よ</b> (It’s OK)		
OK/だ/よ	OK/だ/よ/	OK/だ/よ
Others (insertion)		
Input: す <b>げ</b> 豪華 (very luxury)		
す <b>げ</b> 豪華	すげ/豪華	す <b>げ</b> 豪華

Table 7: Examples of different outputs by introducing derivation rules for long sound symbol substitution and insertion.

text, and thus, there are many likely candidates.

Table 7 lists some examples of the changed output by introducing the derivation rules for informal spelling with long sound symbols. We labeled the change of the analysis “OK だよ” (It’s OK) as negative because the baseline system correctly tagged the POS of “だ” unlike our proposed system, but the baseline system could not also correctly resolve the entire phrase. There was no different output that our proposed system could not resolve but the baseline system could fully resolve.

Table 8 lists some examples of the changed outputs by introducing the derivation rules for informal spelling with lowercase. We labeled the change of the analysis “ゆみ**い**の布団” (Yumi’s bedclothes) as negative because the baseline system correctly segmented the postpositional particle “の” unlike our proposed system. Again for this example, the baseline system could not correctly resolve the entire phrase. Along with the informal spelling with long sound symbols, there was no different output that our proposed system could not resolve but the baseline system could fully resolve.

Rules/patterns	$P_{100D}$	$N_{100D}$	$D_{100kS}$	$P_{100kS}^*$	$N_{100kS}^*$	$Node_{inc.}$	$SP_{loss}$
<i>Rendaku</i> (sequential voicing)	37	8	379	140	30	0.553%	2.0%
Substitution with long sound symbols	55	1	920	506	9	0.048%	0.8%
Substitution with lowercases	78	1	1,762	1,374	18	0.039%	0.7%
Insertion of long sound symbols	84	0	1,301	1,093	0	0.038%	1.9%
Insertion of lowercases	88	0	403	354	0	0.019%	0.4%
Onomatopoeia with repetition	74	2	1,162	860	23	0.021%	0.4%
Onomatopoeia w/o repetition	93	0	250	233	0	0.008%	0.0%
Total	-	-	6,177	4,560	80	0.724%	6.2%

Table 5: Comparison between the analyses with and without the rules/patterns.

Our system	Baseline	Gold standard
Positive (insertion)		
Input: 出して/くれい (please publish)		
出して/くれい	出して/くれい	出して/くれい
Positive (substitution)		
Input: おにいちゃん (big brother)		
おにいちゃん	おにいちゃん	おにいちゃん
Negative (substitution)		
Input: ゆみいの/布団 (Yumi's bedclothes)		
ゆみいの/布団	ゆみいの/布団	ゆみいの/布団
Others (insertion)		
Input: さみすい (lonely)		
さみすい	さみすい	さみすい

Table 8: Examples of different outputs by introducing derivation rules for lowercase substitution and insertion.

Our system	Baseline	Gold standard
Positive		
Input: たゆたゆと (wavy)		
たゆたゆと	たゆたゆと	たゆたゆと
Negative		
Input: あらあら (wow wow)		
あらあら	あら/あら	あら/あら

Table 9: Examples of different outputs by introducing onomatopoeia patterns with repetition.

Our system	Baseline	Gold standard
Positive		
Input: ぺっ/ちゃり (flat)		
ぺっ/ちゃり	ぺっ/ちゃり	ぺっ/ちゃり
Input: チラっと (at a glance)		
チラっと	チラ/っと	チラ/っと

Table 10: Examples of different outputs by introducing onomatopoeia patterns without repetition.

Table 9 lists some examples of the changed outputs by introducing onomatopoeia patterns with repetition. Our system recognized unknown onomatopoeias with repetition at a recall of 89%, which is not very high. However, since there were several repetition expressions other than onomatopoeias, such as “あら/あら” (wow wow) as shown in Table 9, we cannot lessen the cost for onomatopoeias with repetition.

Table 10 lists some examples of the changed outputs by introducing onomatopoeia patterns without repetition. Our system recognized the unknown onomatopoeias without repetition at a recall of 94% and did not output anything worse than

Type	# of types	# of tokens
Covered by Murawaki's Lexicon	13	51
Covered by Wikipedia	68	407
Covered by our method	15	105
Others	22	82
Total	118	645

Table 11: Classification results of unknown words that occur more than two times in KNB corpus.

the baseline output with no loss in speed.

In order to approximate the practical coverage of our method, we classified unknown words that occur more than two times in the Kyoto University and NTT Blog (KNB) corpus<sup>5</sup> into four types: words that are covered by the lexicon created by Murawaki and Kurohashi (2008) (Murawaki's Lexicon), words that are not covered by Murawaki's Lexicon but have entries in Wikipedia, words that are covered only by our method, and the others. Table 11 shows the results. There are total 645 tokens of unknown words that occur more than two times in KNB corpus, 105 of which are newly covered by our method. Since the number of tokens that are covered by neither Murawaki's Lexicon nor Wikipedia is only 187, we can say that the coverage of our method is not trivial.

## 5 Conclusion

We presented a simple approach to unknown word processing in Japanese morphological analysis. Our approach introduced derivation rules and onomatopoeia patterns, and correctly recognized certain types of unknown words. Our experimental results on Web text revealed that our approach could recognize about 4,500 unknown words for 100,000 Web sentences with only 80 harmful side effects and a 6% loss in speed. We plan to apply our approach to machine learning-based morphological analyzers, such as MeCab, with UniDic dictionary, which handles orthographic and phonological variations, in future work.

<sup>5</sup>The KNB corpus consists 4,186 sentences from Japanese blogs, and is available at <http://nlp.kuee.kyoto-u.ac.jp/kunt/>.

## References

- Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proc. of COLING'00*, pages 21–27.
- Masayuki Asahara and Yuji Matsumoto. 2004. Japanese unknown word identification by character-based chunking. In *Proc. of COLING'04*, pages 459–465.
- Ai Azuma, Masayuki Asahara, and Yuji Matsumoto. 2006. Japanese unknown word processing using conditional random fields (in Japanese). In *Proc. of IPSJ SIG Notes NL-173-11*, pages 67–74.
- Richard Beaufort, Sophie Roekhaut, Louise-Amélie Coughon, and Cédric Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proc. of ACL'10*, pages 770–779.
- Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. 2008. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In *Proc. of LREC'08*, pages 1019–1024.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proc. of EMNLP-CoNLL'12*, pages 421–432.
- Kazushi Ikeda, Tadashi Yanagihara, Kazunori Matsumoto, and Yasuhiro Takishima. 2009. Unsupervised text normalization approach for morphological analysis of blog documents. In *Proc. of Australasian Conference on Artificial Intelligence*, pages 401–411.
- Gary Kacmarcik, Chris Brockett, and Hisami Suzuki. 2000. Robust segmentation of japanese text into a lattice for parsing. In *Proc. of COLING'00*, pages 390–396.
- Jun'ichi Kazama, Yutaka Mitsuishi, Makino Takaki, Kentaro Torisawa, Koich Matsuda, and Jun'ichi Tsujii. 1999. Morphological analysis for japanese web chat (in Japanese). In *Proc. of 5th Annual Meetings of the Japanese Association for Natural Language Processing*, pages 509–512.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proc. of EMNLP'04*, pages 230–237.
- Taku Kudo, 2006. *MeCab: Yet Another Part-of-Speech and Morphological Analyzer*. <http://mecab.sourceforge.jp/>.
- Sadao Kurohashi and Daisuke Kawahara. 2005. Japanese morphological analysis system JUMAN version 5.1 manual.
- Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, , and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proc. of The International Workshop on Sharable Natural Language Resources*, pages 22–38.
- Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proc. of ACL'12*, pages 1035–1044.
- Benjamin Smith Lyman. 1894. *The change from surd to sonant in Japanese compounds*. Philadelphia : Oriental Club of Philadelphia.
- Yuji Matsumoto, Kazuma Takaoka, and Masayuki Asahara. 2007. Chasen: Morphological analyzer version 2.4.0 user's manual.
- Shinsuke Mori and Makoto Nagao. 1996. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proc. of COLING'96*, pages 1119–1122.
- Yugo Murawaki and Sadao Kurohashi. 2008. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Proc. of EMNLP'08*, pages 429–437.
- Masaaki Nagata. 1999. A part of speech estimation method for japanese unknown words using a statistical model of morphology and context. In *Proc. of ACL'99*, pages 277–284.
- Tetsuji Nakagawa and Kiyotaka Uchimoto. 2007. A hybrid approach to word segmentation and pos tagging. In *Proc. of ACL'07*, pages 217–220.
- Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. Tsubaki: An open search engine infrastructure for developing new information access methodology. In *Proc. of IJCNLP'08*, pages 189–196.
- Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. The unknown word problem: a morphological analysis of japanese using maximum entropy aided by a dictionary. In *Proc. of EMNLP'01*, pages 91–99.

# Chinese Word Segmentation by Mining Maximized Substrings

Mo Shen, Daisuke Kawahara, and Sadao Kurohashi

Graduate School of Informatics, Kyoto University

Yoshida-honmachi, Sakyo-ku,

Kyoto, 606-8501, Japan

shen@nlp.ist.i.kyoto-u.ac.jp {dk,kuro}@i.kyoto-u.ac.jp

## Abstract

A major problem in the field of Chinese word segmentation is the identification of out-of-vocabulary words. We propose a simple yet effective approach for extracting maximized substrings, which provide good estimations of unknown word boundaries. We also develop a new semi-supervised segmentation technique that incorporates retrieved substrings using discriminative learning. The effectiveness of this novel approach is demonstrated through experiments using both in-domain and out-of-domain data.

## 1. Introduction

Chinese sentences are written without explicit word boundaries, which makes Chinese word segmentation (CWS) an initial and important step in Chinese language processing. Recent advances in machine learning techniques have boosted the performance of CWS systems. On the other hand, a major difficulty in CWS is the problem of identifying out-of-vocabulary (OOV) words, as the Chinese language is continually and rapidly evolving, particularly with the rapid growth of the internet.

A recent line of research to overcome this difficulty is through exploiting characteristics of frequent substrings in unlabeled data. Statistical criteria for measuring the likelihood of a substring being a word have been proposed in previous studies of unsupervised segmentation, such as *accessor variety* (Feng et al., 2004) and *branching entropy* (Jin and Tanaka-Ishii, 2006). This kind of criteria has been applied to enhance the performance of supervised segmentation systems (Zhao and Kit, 2007; Zhao and Kit, 2008;

Substring	Freq
一致	3
界限数的期望值	2
一致认定界限	2
的期望值	3
认定界限数的	2
值	4

Table 1. A particular type of substrings with multiple occurrences in the Chinese sentence: “使一致认定界限数的期望值近似于一致正确界限数的期望值，求得一致认定界限的期望值/认定界限数的值。”

Sun and Xu, 2011) by identifying unknown word boundaries.

In this paper, instead of investigating statistical characteristics of batched substrings, we propose a novel method that extracts substrings as reliable word boundary estimations. The technique uses large-scale unlabeled data, and processes it on the fly.

To illustrate the idea, we first consider the following example taken from a scientific text:

“使一致认定界限数的期望值近似于一致正确界限数的期望值，求得一致认定界限的期望值/认定界限数的值。”

Without any knowledge of the Chinese language one may still notice that some substrings like “一致” and “的期望值”, occur multiple times in the sentence and are likely to be valid words or chains of words. Consider a particular type of frequent substring that cannot be simultaneously extended by its surrounding characters while still being equal (Table 1). We can observe that the boundaries of such substrings can be used as perfect word delimiters. We can segment the sentence by simply treating the boundaries of each occurrence of a substring in Table 1 as word



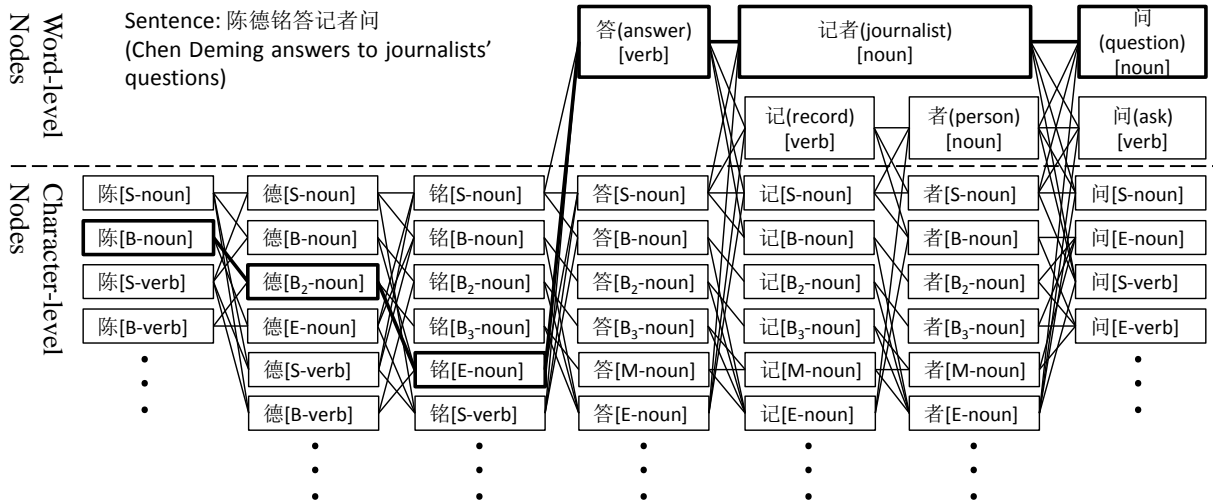


Figure 1. A Word-character hybrid lattice of a Chinese sentence. Correct path is represented by bold lines.

Word Length	1	2	3	4	5	6	7 or more
Tags	<i>S</i>	<i>BE</i>	<i>BB<sub>2</sub>E</i>	<i>BB<sub>2</sub>B<sub>3</sub>E</i>	<i>BB<sub>2</sub>B<sub>3</sub>ME</i>	<i>BB<sub>2</sub>B<sub>3</sub>MME</i>	<i>BB<sub>2</sub>B<sub>3</sub>M...ME</i>

Table 2. Word representation with a 6-tag tagset: *S, B, B<sub>2</sub>, B<sub>3</sub>, M, E*

delimiters:

“使|一致|认定|界限|数|的|期望|值|近似于|一致|正确|界限|数|的|期望|值|，求得|一致|认定|界限|的|期望|值|/|认定|界限|数|的|值|。”

Compared with the gold-standard segmentation, this partial segmentation has a precision of 100% and a recall of 73.3% with regard to boundary estimation. This is high when we consider that the method does not use a trained segmenter or annotated data. While we have obtained this result on a selected instance, it still suggests that unlabeled data has the potential to enhance the performance of supervised segmentation systems by tracking consistency among substrings.

Substrings, such as those listed in Table 1, are retrievable from unlabeled data and can be incorporated with a supervised CWS system to compensate for out-of-vocabulary (OOV) words. In this case the unlabeled data can be either test data only (leading to a purely supervised system), or a large-scale external corpus (leading to a semi-supervised system). We will formally define this particular type of substring, referred to as a “maximized substring”, in a later section.

The remainder of this paper is organized as follows. Section 2 describes our baseline segmentation system, defines maximized substrings, and proposes an efficient algorithm for retrieving these substrings from unlabeled data. Section 3

introduces the maximized substring features. Section 4 presents the experimental results. Section 5 discusses related work. The final section summarizes our conclusions.

## 2. Approach

### 2.1 Baseline Segmentation System

We have used a word-character hybrid model as our baseline Chinese word segmentation system (Nakagawa and Uchimoto, 2007; Kruengkrai et al., 2009). As shown in Figure 1, this hybrid model constructs a lattice that consists of word-level and character-level nodes from a given input sentence. Word-level nodes correspond to words found in the system’s lexicon, which has been compiled from training data. Character-level nodes have special tags called position-of-character (POC) that indicate the word-internal position (Asahara, 2003; Nakagawa, 2004). We have adopted the 6-tag tagset, which (Zhao et al., 2006) reported to be optimal. This tagset is illustrated in Table 2.

Previous studies have shown that jointly processing word segmentation and part-of-speech tagging is preferable to separate processing, which can propagate errors (Nakagawa and Uchimoto, 2007; Kruengkrai et al., 2009). If the training data was annotated by part-of-speech tags, we have combined them with both word-level and character-level nodes.

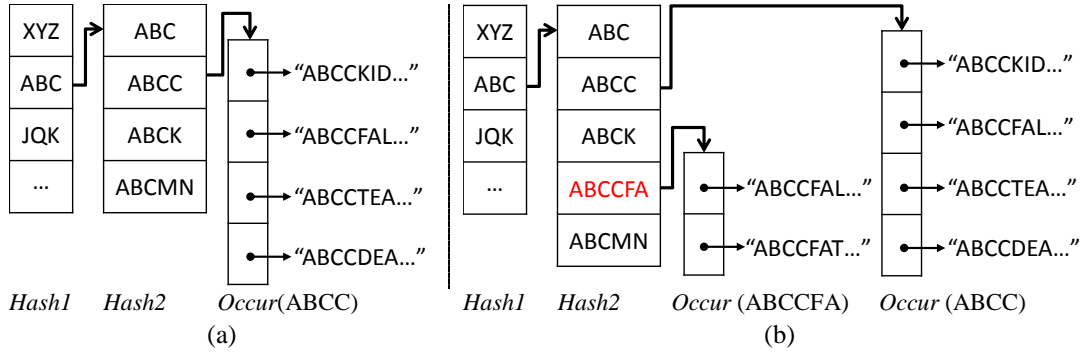


Figure 2. Data structure for maximized substring mining. *Hash1* is the first-level hash with fixed-length prefix keys. *Hash2* is a hash associating to a corresponding key in *Hash1* that stores the list of *maximized substrings* sharing the same fixed-length prefix. *Occur*( $\cdot$ ) is the occurrence list associating to a particular *maximized substrings* with references to all its occurrences in the original positions in the document. (a) shows a certain state of the data structure, and (b) the state after a *maximized substring* “ABCCFA” is inserted with the context being “ABCCFAT...” in the document.

## 2.2 Maximized Substring: the Definition

Frequent substrings in unlabeled data can be used as clues for identifying word boundaries, as we have illustrated in Section 1. Nevertheless, some substrings, although frequent, are not useful to the system. In the example in Section 1, the substring “致认定界” occurs the same amount of times as the substring “一致认定界限”. However, only the latter is a valid identifier for word delimiters: they are non-overlapping, meaning that it is impossible to simultaneously extend all occurrences by surrounding characters. We use the term *maximized substring* to describe these substrings.

Formally, we define *maximized substring* as follows:

*Definition 1 (Maximised substring)*. Given a document  $D$  that is a collection of sentences, denote a length  $n$  substring which starts with character  $c_t$  by  $s_t = [c_t c_{t+1} \dots c_{t+n-1}]$ .  $s_t$  is called a *maximized substring* if:

1. It has a set of distinct occurrences,  $M$ , with at least two elements<sup>1</sup>:  

$$M = \{s_{t_1}, s_{t_2}, \dots, s_{t_m}\}, \quad m > 1, \quad t_1 \neq t_2 \neq \dots \neq t_m \text{ s.t. } s_{t_1} = s_{t_2} = \dots = s_{t_m};$$
 and
2.  $c_{t_i-1} \neq c_{t_j-1}$  and  $c_{t_i+n} \neq c_{t_j+n} \quad \forall i, j = 1, 2, \dots, m, i \neq j$ .

<sup>1</sup> It should be noted that, in order to retrieve a substring, the size of  $M$  is not necessarily identical to its total count in the document.

The substrings listed in Table 1 are therefore maximized substrings, given that  $D$  is the example sentence. Note that these are *not all* maximized substrings extractable from the example sentence, but are the result of the retrieval algorithm that we will describe in the next section.

## 2.3 Maximized Substring Retrieval: Algorithm and Data Structure

The problem of mining frequent substrings in a document has been extensively researched. Existing algorithms generally either use a suffix tree structure (Nelson, 1996) or suffix arrays (Fischer et al., 2005), and make use of the apriori property (Agrawal and Srikant, 1994). The apriori property states that a string of length  $k+1$  is frequent only if its substring of length  $k$  is frequent. The apriori property can significantly reduce the size of enumerable substring candidates. However, as we are only interested in maximized substrings, suffix tree-based algorithms are inefficient in both time and space. We therefore propose a novel algorithm and a compact data structure for fast maximized substring mining.

The data structure is illustrated in Figure 2. It supports fast prefix searching for storing and retrieving maximized substrings, with each entry associated to a list of occurrences that refer to the original positions in the document. Fast prefix matching is a particular advantage of a trie, which is a type of prefix tree. Our structure is different as we use a two-level hash structure for space efficiency and ease of manipulation. This is important, especially during experiments on large-scale unlabeled data.

The first-level hash stores prefixes of a fixed-length,  $m$ , of retrieved substrings. This part of the data structure functions as a filter to screen

out substrings that are shorter than  $m$  characters, as they should not be considered as candidates. This is motivated by our observation that single characters, and sometimes even double-character substrings, are not reliable enough to predict word delimiters. Note that  $m$  is data dependent, for example, the optimal value of  $m$  is 3 characters on the dataset Chinese Treebank (CTB).

Each key of the first-level hash is associated with a second-level hash that stores the retrieved maximized substrings that share a common prefix.

The third-level structure is a linked list of occurrences of a particular maximized substring. This list stores references to the original position of each occurrence of the substring, with the surrounding context being visible so that new (longer) maximized substrings can be found by extension.

We sketch the process of maximized substring retrieval in Pseudocode 1. From the beginning of the document  $D$ , we scan each position and register maximized substrings into the data structure  $H$ . If an incoming substring already exists in  $H$ , we look up its occurrence list to check if its succeeding characters can extend the substring. As the current occurrence list is a set of maximized substrings, there will be only two possible outcomes. Either exactly one element in the occurrence list is found to have a longer common prefix with the incoming substring, in which case we create a new occurrence list consisting of the two lengthened substrings. Alternatively, the prefix remains the same and we add the incoming substring to the occurrence list.

We can easily demonstrate that all substrings retrieved by this algorithm are maximized substrings. However, the algorithm does not generally guarantee to retrieve all maximized substrings from unlabeled data. This is a necessary compromise if we wish to keep the efficiency of one-time scanning. In addition, we have observed in preliminary experiments that retrieving all maximized substrings is not only unnecessary, but can introduce harmful noise. In the next section, we will discuss our solution to this problem.

## 2.4 Short-Term Store

Maximized substrings can provide good estimations of word boundaries, but random noise can be introduced during the retrieval process in Pseudocode 1.

To address this problem, we take advantage of a linguistic phenomenon. It has been observed that a word occurring in the recent past has a

---

### Pseudocode 1: Maximized substring retrieval

---

```

1  procedure RetrieveMaxSub( $m, D$ )
2     $i \leftarrow 0, H \leftarrow \emptyset$ 
3     $p \leftarrow [c_0 c_1 \dots c_{m-1}]$ 
4     $\triangleleft$  the reference of a length  $m$  substring at
5    the beginning of document  $D$ 
6    until  $i$  reaches the end of document  $D$ 
7       $s \leftarrow$  longest element in  $H$  extendable
8      from  $p$ 
9      if  $|s| = 0$   $\triangleleft$  empty string
10        $occurList \leftarrow \{p\}$ 
11        $\triangleleft$  make the occurrence list of  $p$ 
12        $H.Add(\langle p, occurList \rangle)$ 
13        $\triangleleft$  associate  $p$  with its occurrence list
14       and add to data structure
15        $i \leftarrow i + 1$ 
16        $p \leftarrow [c_i \dots c_{i+m-1}]$ 
17     else
18        $p^* \leftarrow [c_i \dots c_{i+|s|-1}]$ 
19        $(i, p) \leftarrow \text{Maximize}(H, m, i, s, p^*)$ 
20     return  $H$ 
21
22  procedure Maximize( $H, m, i, s, p^*$ )
23    for each  $e$  in  $s.occurList$ 
24       $\langle s_{new}, e_{new}, p_{new}^* \rangle \leftarrow \text{Extend}(e, p^*)$ 
25       $\triangleleft$  find the longest common substring
26       $s_{new}$  between  $e$  and  $p^*$  by simultane-
27      ously extending them with succeeding
28      characters
29      if  $|s_{new}| > |s|$ 
30         $occurList_{new} \leftarrow \{e_{new}, p_{new}^*\}$ 
31         $H.Add(\langle s_{new}, occurList_{new} \rangle)$ 
32         $i \leftarrow i + |s_{new}|$ 
33         $p \leftarrow [c_i \dots c_{i+m-1}]$ 
34      return  $(i, p)$ 
35  end
36   $s.occurList.Add(p^*)$ 
37   $i \leftarrow i + |s|$ 
38   $p \leftarrow [c_i \dots c_{i+m-1}]$ 
39  return  $(i, p)$ 

```

---

much higher probability to occur again soon, when compared with its overall frequency (Kuhn and Mori, 1990). It follows that, for speech recognition, we can then use a window of recent history to adjust the static overall language mode.

This observation is applicable to the task of maximized substring retrieval in the following way. Suppose a substring is registered into the data structure. If the substring is in fact a word, it is much more likely to reoccur in the next 50 to 100 sentences than in the remainder of the corpus (especially when it is a technical term or a named entity). Otherwise the substring should have a more unified probability of reoccurrence across the entire corpus.

This motivated us to introduce a functionality into the process of maximized substring retrieval, called “short-term store” (STS). The STS is an analogy to the cache component in speech recognition as well as the human phonological working memory in language acquisition. It restricts the length of the visible context when retrieving the next candidate of a registered substring, making it proportional to the current number of occurrences of the substring. For a registered substring, the retrieval algorithm scans a certain number of sentences after the latest occurrence of the substring, where the number of sentences  $D(s)$  is determined as follows:

$$D(s) = \begin{cases} \lambda \cdot \text{count}(s), & \text{if } \text{count}(s) < \theta, \\ \infty, & \text{otherwise,} \end{cases}$$

where  $\text{count}(s)$  is the current number of occurrences of  $s$  in the data structure. The parameter  $\lambda$  contributes a fixed-length distance to the visible context. The parameter  $\theta$  works as a threshold of reliability. If we have observed  $s$  at least  $\theta$  times in a short period, we can regard  $s$  as a word, or a sequence of words, with a high level of confidence. Thus,  $D(s) = \infty$  implies that  $s$  is no longer subject to periodical decaying and will stay in the data structure statically.

During the scanning of the  $D(s)$  sentences, if a new occurrence of  $s$  is found, it is added into the data structure and  $D(s)$  is recalculated immediately, starting a new scanning period. If no new occurrences are found, we remove the earliest occurrence of  $s$  from the data structure and then re-calculate  $D(s)$ . Note that we have described the short-term store functionality as if each substring in the data structure is scanned separately. In practice, however, only a small change to Pseudocode 1 is required so that STS is used, making one-time scanning of the unlabeled data sufficient.

Introducing STS into the retrieval process results in a substantial improvement to the quality of retrieved substrings. It is also important that STS greatly improves the processing efficiency for large scale unlabeled data by keeping the size of the data structure relatively small. This is because a substring entry will decay from the data structure if it has not been refreshed in a short period.

### 3. Features

#### 3.1 Baseline Features

For baseline features, we apply the feature tem-

plates described in (Kruengkrai et al., 2009). For further details, please see the original paper. Note that if the part-of-speech tags are not available, we omit those templates involving POS tags.

#### 3.2 Maximized Substring Features

We have incorporated the list of retrieved maximized substrings into the baseline system by using a technique which discriminatively learns their features. For every word-level and character-level node in the lattice, the method checks the maximized substring list for entries that satisfy the following two conditions:

1. The node matches the maximized substring at the beginning, the end, or both boundaries.
2. The length of the node is shorter than or equal to that of the entry.

For example, consider the lattice in Figure 1 with a maximized substring “陈德铭”. All of the character-level nodes of “陈” and “铭” are encoded with maximized substring features. A segmenter will only obtain information on those possible word boundaries that are identified by maximized substrings. The maximized substrings are not directly treated as single words, because a maximized substring can sometimes be a compound word or phrase.

For each match with a maximized substring entry, the technique encodes the following features.

**Basic:** A binary feature that indicates whether the match is at the beginning or end of the maximized substring. It is encoded both individually and as a combination with each other feature types.

**Lexicon:** There is a particular kind of noise in the retrieved list of maximized substrings, namely, those like the substring “中美经”, which has resulted from the two phrases “中美经济” (China and U.S. economy) and “中美经贸” (China and U.S. economic and trade). This happens when the boundary of a maximized substring is a shared boundary character of multiple other words. In this example, the last character “经” of the maximized substring is the character at the beginning of “经济” (economy) and “经贸” (economic and trade). This kind of noise can be identified by checking the context of maximized substrings in system’s lexicon.

Our technique checks the context of the maximized substring in the input sentence and compares it with the system’s lexicon. If any item in the lexicon is found that forms a positional relation with the maximized substring entry (as listed

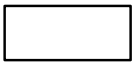

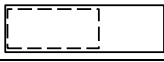
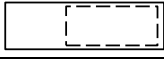

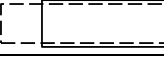

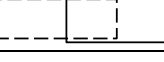


Sentence: $s = \dots c_{-1}c_0c_1 \dots c_{n-1}c_nc_{n+1} \dots$		Representa- tion
Maximized substring $m = c_0c_1c_2 \dots c_n$		
Lexicon entry $l = c_ic_{i+1}c_{i+2} \dots c_j$		
ID	Positional Relation	
L1	$0 = i < j < n$	
L2	$0 < i < j = n$	
L3	$0 = i < n < j$	
L4	$i < 0 < j = n$	
L5	$0 < i < n < j$	
L6	$i < 0 < j < n$	
L7	$i = n + 1$	
L8	$j = -1$	

Table 3. Lexicon features. Each one represents a positional relation between a maximized substring and a contextual substring which exists in system’s lexicon.

ID	At Beginning	ID	At Ending
B1	<L1,L6>	E1	<L2,L5>
B2	<L6,L8>	E2	<L5,L7>
B3	<L1,L8>	E3	<L2,L7>

Table 4. Lexicon Composition features. Each one represents a combination of two Lexicon features that fire simultaneously.

in Table 3) then the corresponding features are encoded.

**Lexicon Composition:** When a maximized substring is a match to more than one item in the lexicon, a combination of multiple lexicon features is more informative than individual features. We encode the combinations of lexicon features listed as in Table 4.

**Frequency:** We sort the list of maximized substrings by their frequencies. If a maximized substring is among the 10% most frequent it is classed as “highly frequent”, if it is among the top 30% it is “normal”, and all other cases are “infrequent”.

## 4. Evaluation

### 4.1 Setting

To evaluate our approach, we have conducted word segmentation experiments on two datasets. The first is Chinese Treebank 7 (CTB7), which is a widely used version of the Penn Chinese Treebank dataset for the evaluations of word segmentation techniques. We have adopted the same setting of data division as (Wang et al., 2011): the training set, dev set and test set. For CTB7, these sets have 31,131, 10,136 and 10,180 sentences respectively. The second dataset is the second international Chinese word segmentation bakeoff (SIGHAN Bakeoff-2005) (Emerson, 2005), which has four independent subsets: the Academia Sinica Corpus (AS), the Microsoft Research Corpus (MSR), the Hong Kong City University Corpus (CityU) and the Peking University Corpus (PKU). Since POS tags are not available in this dataset, we have omitted all templates that include them. The models and parameters applied on all test sets are those that result in the best performance on the CTB7 dev set.

We have used two different types of unlabeled data. One is the test set itself, which means the system is purely supervised. Another is a large-scale dataset, which is the Chinese Gigaword Second Edition (LDC2007T03). This dataset is a collection of news articles from 1991 to 2004 published by Central News Agency (Taiwan), Xinhua News Agency and Lianhe Zaobao Newspaper. It includes a total amount of over 1.2 billion characters in both simplified Chinese and traditional Chinese.

We have trained all models using the averaged perceptron algorithm (Collins, 2002), which we selected because of its efficiency and stability. To learn the characteristics of unknown words, we built the system’s lexicon using only the words in the training data with a frequency higher than a threshold,  $h$ . This threshold was tuned using the development data. In order to use the maximized substring features, we have used training data as unlabeled data for supervised models, and used both the training data and Chinese Gigaword for semi-supervised models.

We have applied the same parameters for all models, which are tuned on the CTB7 dev set:  $m = 3$ ,  $h = 2$ ,  $\lambda = 100$ , and  $\theta = 3$ .

We have used precision, recall and the F-score to measure the performance of segmentation systems. Precision,  $p$ , is defined as the percentage of

System	P	R	F
Baseline	95.17	95.35	95.26
MaxSub-Test	95.33	95.47	95.40
MaxSub-U	95.65	95.81	95.73

Table 5. Evaluation on CTB7 for the baseline approach and our approach with small and large-scale in-domain unlabeled data respectively.

words that are segmented correctly, and recall,  $r$ , is the percentage of words in the gold standard data that are recognized in the output. The balanced F-score is defined as  $F = 2pr/(p + r)$ .

## 4.2 Experimental Results on In-domain Data

We have compared the performance between the baseline system and our approach. The results are shown in Table 5. Each row in this table shows the performance of the corresponding system. “Baseline” refers to our baseline hybrid word segmentation and POS-tagging system. “MaxSub-Test” refers to the method that just uses the test set as unlabeled data. “MaxSub-U” refers to the method that uses the large-scale unlabeled data. We have focused on the segmentation performance of our systems.

The results show that, using the test data as an additional source of information, “MaxSub-Test” outperforms the baseline method by 0.14 points in F-score. This indicates that our method of using maximized substrings can enhance the segmentation performance even with a purely supervised approach. The improvement increases to 0.47 points in F-score for “MaxSub-U”, which demonstrates the effectiveness of using large-scale unlabeled data.

We have compared our approach with previous work in Table 6. Two methods from (Kruengkrai et al., 2009a; 2009b) are referred to as “Kruengkrai 09a” and “Kruengkrai 09b”, and are taken directly from the report of (Wang et al., 2011). “Wang 11” refers to the semi-supervised system in (Wang et al., 2011). We have observed that our system “MaxSub-U” achieves the best segmentation among these systems. Also, although the performance of our baseline is lower than the systems “Kruengkrai 09a” and “Kruengkrai 09b” because of differences in implementation, the system “MaxSub-Test” (which has used no external resource) has achieved a comparable result.

The results for the SIGHAN Bakeoff-2005 dataset are shown in Table 7. The first three rows (“Tseng 05”, “Asahara 05” and “Chen 05”) show the results of systems that have reached the highest score on at least one corpus (Tseng et al.,

System	F
Baseline	95.26
MaxSub-Test	95.40
MaxSub-U <sup>+</sup>	<b>95.73</b>
Kruengkrai 09a	95.40
Kruengkrai 09b	95.46
Wang 11 <sup>+</sup>	95.65

Table 6. F-measure on CTB7 test set compared with previous work. “<sup>+</sup>”: semi-supervised systems.

System	AS	CityU	MSR	PKU
Tseng 05	94.7	<b>94.3</b>	<b>96.4</b>	<b>95.0</b>
Asahara 05	<b>95.2</b>	94.1	95.8	94.1
Chen 05	94.5	94.0	96.0	<b>95.0</b>
Best closed	95.2	94.3	96.4	95.0
Zhang 07	95.1	95.1	97.2	95.1
Zhao 07	95.5	95.6	97.5	95.4
Baseline	95.07	94.53	96.25	95.13
MaxSub-S	95.17	94.61	96.42	95.31
MaxSub-L <sup>+</sup>	95.34	94.79	96.64	<b>95.55</b>

Table 7. F-measure on SIGHAN Bakeoff-2005 test set compared with previous work. “<sup>+</sup>”: semi-supervised systems.

2005; Asahara et al., 2005; Chen et al., 2005). “Best closed” summarizes the best official results on all four corpora. “Zhao 07” and “Zhang 06” represent the supervised segmentation systems in (Zhao and Kit, 2007; Zhang et al., 2006). “Baseline”, “Maxsub-Test” and “MaxSub-U” refer to the same systems as in Table 5. For the unlabeled data, we have used the test sets of corresponding corpora for “MaxSub-Test”, and the Chinese Gigaword for “MaxSub-U”. Other parameters were left unchanged. The results do not indicate that our approach performs better than other systems. However, this is largely because of our baseline not being optimized for these corpora. Nevertheless, when compared with the baseline, our approach has yielded consistent improvements across the four corpora, and on the PKU corpus we have performed better than previous work.

## 4.3 Impacts of Semi-supervised Features and Short-term Store

In Table 8, we have shown the effects of the different maximized substring feature types proposed in this paper. We activated different combinations of feature types in turn and trained separate models. We also investigated the impact of the short-term store by training models without this feature. The rows of this table represent models and corresponding F-measure, trained

System	F
Baseline	95.26
+Basic&Freq	95.50
+All	95.60
+All+STS	<b>95.73</b>

Table 8. Influence of activated feature types and short-term store on CTB7 test data.

System	P	R	F
Baseline	91.88	92.02	91.95
MaxSub-Test	92.43	92.53	92.48

Table 9. Results on out-of-domain data.

and tested on CTB7 with different configurations. The row “Baseline” is baseline system as in Table 5. “+Basic&Freq” represents the system “MaxSub-U” with only basic and frequency features activated, and STS turned off. The row “+All” represents a system activating all maximized substring features but still without STS. The last row “+All+STS” is identical to the system “Maxsub-U”. It is clear that lexicon-based features are effective in discriminating unreliable maximized substring from reliable ones, and the short-term store improves the segmentation performance by filtering out noises during the retrieval of maximized substrings. The combination of these two techniques yields an improvement of 0.23 point in F-measure, and thus are essential when using maximized substrings.

#### 4.4 Experimental Results on Out-of-domain Data

To demonstrate the effectiveness of our method on out-of-domain text, we have conducted an experiment on a test set that was drawn from a corpus of scientific articles. This test set contains 510 sentences that have been manually segmented by a native Chinese speaker. We used the test set as the unlabeled data.

As the results show (Table 9), the system “MaxSub-Test” exceeded the baseline method by 0.53 in F-score, which is a significant improvement. Considering that the amount of unlabeled data is relatively small, it is likely that acquiring large-scale unlabeled data in the same domain will further benefit the accuracy.

## 5. Related Work

The authors of (Feng et al., 2004) proposed accessor variety (AV), a criterion measuring the likelihood of a substring being a word by count-

ing distinct surrounding characters. In (Jin and Tanaka-Ishii, 2006) the researchers proposed branching entropy, a similar criterion based on the assumption that the uncertainty of surrounding characters of a substring peaks at the word boundaries. The authors of (Zhao and Kit, 2007) incorporated accessor variety and another type of criteria, called co-occurrence sub-sequence, with a supervised segmentation system and conducted comprehensive experiments to investigate their impacts. Although the idea behind co-occurrence sub-sequence is similar with maximized substrings, there are several restrictions: it requires post-processing to remove overlapping instances; sub-sequences are retrievable only from different sentences; and the retrieval is performed only on training and testing data. In (Sun and Xu, 2011), the authors proposed a semi-supervised segmentation system enhanced with multiple statistical criteria. Large-scale unlabeled data were used in their experiments.

Li and Sun presented a model to learn features of word delimiters from punctuation marks in (Li and Sun, 2009). Wang et al. proposed a semi-supervised word segmentation method that took advantages from auto-analyzed data (Wang et al., 2011).

Nakagawa showed the advantage of the hybrid model combining both character-level information and word-level information in Chinese and Japanese word segmentation (Nakagawa, 2004). In (Nakagawa and Uchimoto, 2007) and (Kruengkrai et al., 2009a; 2009b) the researchers presented word-character hybrid models for joint word segmentation and POS tagging, and achieved the state-of-the-art accuracy on Chinese and Japanese datasets.

## 6. Conclusion

We propose a simple yet effective approach for extracting maximized substrings from unlabeled data. These are a particular type of substrings that provide good estimations of unknown word boundaries. The retrieved maximized substrings are incorporated with a supervised segmentation system through discriminative learning. We have demonstrated the effectiveness of our approach through experiments in both in-domain and out-of-domain data and have achieved significant improvements over the baseline systems across all datasets<sup>2</sup>.

<sup>2</sup>  $p < 0.05$  in McNemar’s test.

## References

- Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules. In Proceedings of 1994 Int. Conf. Very Large Data Bases, pages 487–499.
- Masayuki Asahara. 2003. Corpus-based Japanese Morphological Analysis. Nara Institute of Science and Technology, Doctor's Thesis.
- Masayuki Asahara, Kenta Fukuoka, Ai Azuma, Chooi-Ling Goh, Yotaro Watanabe, Yuji Matsumoto, and Takashi Tsuzuki. 2005. Combination of Machine Learning Methods for Optimum Chinese Word Segmentation. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 134–137.
- Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In Proceedings of EMNLP 2002, pages 1–8.
- Thomas Emerson. 2005. The Second International Chinese Word Segmentation Bakeoff. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 123–133.
- Aitao Chen, Yiping Zhou, Anne Zhang, and Gordon Sun. 2005. Unigram language model for Chinese word segmentation. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 138–141.
- Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30(1), pages 75–93.
- Johannes Fischer, Volker Heun, and Stefan Kramer. 2005. Fast Frequent String Mining Using Suffix Arrays. In Proceedings of ICDM 2005, IEEE Computer Society, pages 609–612.
- Zhihui Jin and Kumiko Tanaka-Ishii. 2006. Unsupervised Segmentation of Chinese Text by Use of Branching Entropy. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 428–435.
- Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, YiuWang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An Error-Driven Word-Character Hybrid Model for Joint Chinese Word Segmentation and POS Tagging. In Proceedings of ACL/IJCNLP 2009, pages 513–521.
- Canasai Kruengkrai Kiyotaka Uchimoto, Jun'ichi Kazama, Yiu Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. Joint Chinese Word Segmentation and POS Tagging Using an Error-Driven Word-Character Hybrid Model. *IEICE transactions on information and systems*, 92(12), pages 2298–2305.
- Roland Kuhn and Renato De Mori. 1990. A Cache-based Natural Language Model for Speech Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(6), pages 570–583.
- Zhongguo Li and Maosong Sun. 2009. Punctuation as Implicit Annotations for Chinese Word Segmentation. *Computational Linguistics*, 35(4), pages 505–512.
- Mark Nelson. 1996. Fast String Searching with Suffix Trees. *Dr.Dobb's Journal*.
- Tetsuji Nakagawa. 2004. Chinese and Japanese Word Segmentation Using Word-level and Character-level Information. In Proceedings of COLING 2004, pages 466–472.
- Tetsuji Nakagawa and Kiyotaka Uchimoto. 2007. Hybrid Approach to Word Segmentation and Pos Tagging. In Proceedings of ACL 2007 Demo and Poster Sessions, pages 217–220.
- Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese Word Segmentation and POS Tagging with Semi-supervised Methods Using Large Auto-Analyzed Data. In Proceedings of IJCNLP 2011.
- Weiwei Sun and Jia Xu. 2011. Enhancing Chinese Word Segmentation Using Unlabeled Data. In Proceedings of EMNLP 2011, pages 970–979.
- Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A Conditional Random Field Word Segmenter for SIGHAN Bakeoff 2005. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pages 168–171.
- Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling. In Proceedings of PACLIC 20, pages 87–94.
- Hai Zhao and Chunyu Kit. 2007. Incorporating Global Information into Supervised Learning for Chinese Word Segmentation. In Proceedings of PACLING 2007, pages 66–74.
- Hai Zhao and Chunyu Kit. 2008. Exploiting Unlabeled Text with Different Unsupervised Segmentation Criteria for Chinese Word Segmentation. *Research in Computing Science*, Vol. 33, pages 93–104.
- Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based Tagging for Confidence Dependent Chinese Word Segmentation. In COLING/ACL 2006, pages 961–968.



# Capturing Long-distance Dependencies in Sequence Models: A Case Study of Chinese Part-of-speech Tagging

Weiwei Sun and Xiaochang Peng and Xiaojun Wan

Institute of Computer Science and Technology, Peking University

The MOE Key Laboratory of Computational Linguistics

{ws, wanxiaojun}@pku.edu.cn; pxc.pku@gmail.com

## Abstract

This paper is concerned with capturing long-distance dependencies in sequence models. We propose a two-step strategy. First, the stacked learning technique is applied to integrate sequence models that are good at exploring local information and other high complexity models that are good at capturing long-distance dependencies. Second, the structure compilation technique is employed to transfer the predictive power of hybrid models to sequence models via large-scale unlabeled data. To investigate the feasibility of our idea, we study Chinese POS tagging. Experiments on the Chinese Treebank data demonstrate the effectiveness of our methods. The re-compiled models not only achieve high accuracy with respect to per token classification, but also serve as a front-end to a parser well.

## 1 Introduction

Sequential classification models provide very important solutions to pattern recognition tasks that involve the automatic assignment of a categorical label to each token of a sequence of observed values. A common example is part-of-speech (POS) tagging, which seeks to assign a grammatical category to each word in an input sentence. *Standard* machine learning algorithms to sequential tagging, e.g. linear-chain conditional random fields and max-margin Markov network, directly exploit local dependencies and perform quite well for a large number of sequence labeling tasks. In these models, usually, the relationships between two (or three) successive labels are parameterized and encoded as a single feature, and Viterbi style dynamic programming algorithms are applied to inference over a lattice. Although sequence models

perform well for many applications, they are inadequate for tasks where many long-distance dependencies are involved.

Sequential classification models play an important role in natural language processing (NLP). Several fundamental NLP tasks, including named entity recognition, POS tagging, text chunking, supertagging, etc., employ sequential classifiers for lexical and syntactic disambiguation. In addition to learning linear chain structures, sequence models can even be applied to acquire hierarchical syntactic structures (Tsuruoka et al., 2009). However, long-distance dependencies widely exist in linguistic structures, and many NLP systems suffer from the incapability of capturing these dependencies. For example, previous work has shown that sequence models alone cannot deal with syntactic ambiguities well (Clark and Curran, 2004; Tsuruoka et al., 2009). On the contrary, state-of-the-art systems usually utilize high complexity models, such as lexicalized PCFG models for syntactic parsing, to achieve high accuracy. Unfortunately, they are not suitable for many real world applications due to the sacrifice of efficiency.

In this paper, we are concerned with capturing long-distance dependencies in sequence models. Our goal is to develop efficient models with linear time complexity that are also capable to capture non-local dependencies. Two techniques are studied to achieve this goal. First, stacked learning (Breiman, 1996) is employed to integrate sequence models that are good at exploring local information and other high complexity models that are good at capturing non-local dependencies. By combining complementary strengths of heterogeneous models, hybrid systems can obtain more accurate results. Second, structure compilation (Liang et al., 2008) is employed to transfer the predictive power of hybrid models to sequence models via large-scale unlabeled data. In particular, hybrid systems are utilized to create large-scale

pseudo training data for cheap sequence models. A discriminative model can be improved by incorporating more features, while a generative latent variable model can be improved by increasing the number of latent variables. By using stacking and structure compilation techniques, a sequence model can be enhanced to better capture long-distance dependencies and to achieve more accurate results.

To demonstrate the feasibility to capture long-distance dependencies in a sequence model, we present our work on Chinese POS tagging. The Chinese language has a number of characteristics that make Chinese POS tagging particularly challenging. While simple sequential classifiers can easily achieve tagging accuracies of above 97% on English, Chinese POS tagging has proven to be more challenging and has obtained accuracies of about 93-94% (Huang et al., 2009; Sun and Uszkoreit, 2012) when applying sequence models. Recent work shows that higher accuracy (c.a. 95%) can be achieved by applying advanced learning techniques to capture deep lexical relations (Sun and Uszkoreit, 2012). Especially, syntagmatic lexical relations have been shown playing an essential role in Chinese POS tagging. To capture such relations, an accurate POS tagging model should know more information about long range dependencies. Previous work has used syntactic parsers in either constituency or dependency formalisms to exploit such useful information (Sun and Uszkoreit, 2012; Hatori et al., 2011). However, it is inappropriate to employ computationally expensive parsers to improve POS tagging for many realistic NLP applications, mainly due to efficiency considerations.

In this paper, we study several hybrid systems that are built upon various complementary tagging systems. We investigate stacked learning to build more accurate solutions by integrating heterogeneous models. Experiments on the Chinese Treebank (CTB) data show that stacking is very effective to build high-accuracy tagging systems. Although predictive powers of hybrid systems are significantly better than individual systems, they are not suitable for large-scale real word applications that have stringent time requirements. To improve POS tagging efficiency without loss of accuracy, we explore unlabeled data to transfer the predictive power of complex, inefficient models to simple, efficient models. Experiments show that

unlabeled data is effective to re-compile simple models, including latent variable hidden Markov models, local and global linear classifiers. On one hand, the precision in terms of word classification is improved to 95.33%, which reaches the state-of-the-art. On the other hand, re-compiled models are adapted based on parsing results, and as a result the ability to capture syntagmatic lexical relations is improved as well. Different from the purely supervised sequence models, re-compiled models also serve as a front-end to a parser well.

## 2 Background

The Chinese language has a number of characteristics that make Chinese POS tagging particularly challenging. For example, Chinese is characterized by the lack of formal devices such as morphological tense and number that often provide important clues for syntactic processing. Chinese POS tagging has proven to be very difficult and has obtained accuracies of about 93-94% (Huang et al., 2009; Li et al., 2011; Hatori et al., 2011; Sun and Uszkoreit, 2012). On the other hand, Chinese POS information is very important for advanced NLP tasks, e.g. supertagging, full parsing and semantic role labeling. Previous work has repeatedly demonstrated the significant performance gap of NLP systems while using gold standard and automatically predicted POS tags (Zhang and Clark, 2009; Li et al., 2011; Tse and Curran, 2012). In this section, we give a brief introduction and a comparative analysis to several models that are recently designed to resolve the Chinese POS tagging problem.

### 2.1 Various Chinese POS Tagging Models

**Local linear model (LLM)** A very simple approach to POS tagging is to formulate it as a local word classification problem. Various features can be drawn upon information sources such as word forms and characters that constitute words. Previous studies on many languages have shown that local classification is inadequate to capture structural information of output labels, and thus does not perform as well as structured models.

**Linear-chain global linear model (LGLM)** Sequence labeling models can capture output structures by exploiting local dependencies among words. A global linear model is flexible to in-

clude linguistic knowledge from multiple information sources, and thus suitable to recognize more new words. A majority of state-of-the-art English POS taggers are based on LGLMs, e.g. structured perceptron (Collins, 2002) and conditional random fields (Lafferty et al., 2001). Such models are also very popular for building Chinese POS taggers (Sun and Uszkoreit, 2012).

**Hidden Markov model with latent variables (HMMLA)** Generative models with latent annotations (LA) obtain state-of-the-art performance for a number of NLP tasks. For example, both PCFG and TSG with refined latent variables achieve excellent results for syntactic parsing (Matsuzaki et al., 2005; Shindo et al., 2012). For Chinese POS tagging, Huang, Eidelman and Harper (2009) described and evaluated a bi-gram HMM tagger that utilizes latent annotations. The use of latent annotations substantially improves the performance of a simple generative bigram tagger, outperforming a trigram HMM tagger with sophisticated smoothing.

**PCFG Parsing with latent variables (PCFGLA)** POS tags can be taken as preterminals of a constituency parse tree, so a constituency parser can also provide POS information. The majority of the state-of-the-art constituent parsers are based on generative PCFG learning, with lexicalized (Collins, 2003; Charniak, 2000) or latent annotation (Matsuzaki et al., 2005; Petrov et al., 2006) refinements. Compared to complex lexicalized parsers, the PCFGLA parsers leverage on an automatic procedure to learn refined grammars and are more robust to parse many non-English languages that are not well studied. For Chinese, a PCFGLA parser achieves the state-of-the-art performance and outperforms many other types of parsers (Zhang and Clark, 2009).

### 2.1.1 Joint POS Tagging and Dependency Parsing (DEP)

(Hatori et al., 2011) proposes an incremental processing model for the task of joint POS tagging and dependency parsing, which is built upon a shift-reduce parsing framework with dynamic programming. Given a segmented sentence, a joint model simultaneously considers possible POS tags and dependency relations. In this way, the learner can better predict POS tags by using bi-lexical dependency information. Their experiments show that the joint approach achieved substantial im-

provements over the pipeline systems in both POS tagging and dependency parsing tasks.

## 2.2 Comparison

We can distinguish the five representative tagging models from two views (see Table 2). From a linguistic view, we can distinguish syntax-free and syntax-based models. In a syntax-based model, POS tagging is integrated into parsing, and thus (to some extent) is capable of capturing long range syntactic information. From a machine learning view, we can distinguish generative and discriminative models. Compared to generative models, discriminative models define expressive features to classify words. Note that the two generative models employ latent variables to refine the output spaces, which significantly boost the accuracy and increase the robustness of simple generative models.

	Generative	Discriminative
Syntax-free	HMMLA	LLM, LGLM
Syntax-based	PCFGLA	DEP

Table 2: Two views of different tagging models.

## 2.3 Evaluation

### 2.3.1 Experimental Setting

Penn Chinese Treebank (CTB) (Xue et al., 2005) is a popular data set to evaluate a number of Chinese NLP tasks, including word segmentation, POS tagging, syntactic parsing in both constituency and dependency formalisms. In this paper, we use CTB 6.0 as the labeled training data for the study. In order to obtain a representative split of data sets, we conduct experiments following the setting of the CoNLL 2009 shared task (Hajič et al., 2009), which is also used by (Sun and Uszkoreit, 2012). The setting is provided by the principal organizer of the CTB project, and has considered many annotation details. This setting is very robust for evaluating Chinese language processing algorithms.

We present an empirical study of the five typical approaches introduced above. In our experiments, to build local and global word classifiers (i.e. LLMs and LGLMs), we implement the feature set used in (Sun and Uszkoreit, 2012). Denote a word  $w$  in focus with a fixed window  $w_{-2}w_{-1}ww_{+1}w_{+2}$ . The features include:

- Word unigrams:  $w_{-2}, w_{-1}, w, w_{+1}, w_{+2}$ ;

Devel.	LLM	LGLM(SP)	LGLM(PA)	HMMLA	PCFGLA	DEP
Overall	93.96%	94.30%/94.49%	94.24%/94.33%	94.16%	93.69%	94.58%
NR	95.07	94.47/94.85	94.41/94.56	94.22	89.84	93.55
NT	97.61	97.22/97.75	97.66/97.59	97.18	96.70	96.84
NN	94.89	94.67/94.79	94.72/94.71	94.30	93.56	94.55
DEC	78.61	81.98/82.36	80.68/81.76	80.60	85.78	86.73
DEG	82.44	85.58/86.72	85.37/85.00	85.19	88.94	89.45
UNK	--	80.0%/81.1%	--	78.2%	--	--

Table 1: Tagging accuracies of different supervised models on the development data.

- Word bigrams:  $w_{-2}w_{-1}$ ,  $w_{-1}w$ ,  $w_w w_{+1}$ ,  $w_{+1}w_{+2}$ ;
- Character  $n$ -gram prefixes and suffixes for  $n$  up to 3.

To train LLMs, we use the open source linear classifier – LIBLINEAR<sup>1</sup>. To train LGLMs, we choose structured perceptron (SP) (Collins, 2002) and passive aggressive (PA) (Crammer et al., 2006) learning algorithms. For the LAHMM and DEP models, we use the systems described in (Huang et al., 2009; Hatori et al., 2011); for the PCFGLA models, we use the Berkeley parser<sup>2</sup>.

### 2.3.2 Results

Table 1 summarizes the performance in terms of per word classification of different supervised models on the development data. We present the results of both first order (on the left) and second order (on the right) LGLMs. We can see that the perceptron algorithm performs a little better than the PA algorithm for Chinese POS tagging. There is only a slight gap between the local classification model and various structured models. This is very different from English POS tagging. Although the local classifier achieves comparable results when respectively applied to English and Chinese, there is much more significant gap between the corresponding structured models. Similarly, the gap between the first and second order LGLMs is very modest too.

From the linguistic view, we mainly consider the disambiguation ability of local and non-local dependencies. Table 1 presents accuracy results of several POS types, including nouns and functional words. The POS types *NR*, *NT* and *NN* respectively represent proper nouns, temporal nouns and other common nouns. We can clearly see that models which only explore local dependencies are

good enough to deal with nouns. Surprisingly, the local classifier that does not directly define features of possible POS tags of other surrounding words performs even better than structured models for proper nouns and other common nouns.

The tag *DEC* denotes a complementizer or a nominalizer, while the tag *DEG* denotes a genitive marker and an associative marker. These two types only include two words: “的” and “之.” The latter one is mainly used in ancient Chinese. 5.19% of words appearing in the training data set is *DEC/DEG*. The pattern of the *DEC* recognition is *clause/verb phrase+DEC+noun phrase*, and The pattern of the *DEG* recognition is *nominal modifier+DEC+noun phrase*. To distinguish the sentential/verbal and nominal modification phrases, the *DEC* and *DEG* words usually need long range syntactic information for accurate disambiguation. We claim that the prediction performance of the two specific types is a good clue of how well a tagging model resolves long distance dependencies. We can see that the two syntactic parsers significantly outperform local models on the prediction of these types of words.

The weak ability for non-local disambiguation also imposes restrictions on using a sequence POS tagging model as front module for parsing. To evaluate the impact, we employ the PCFGLA parser to parse a sentence based on the POS tags provided by sequence models. Table 4 shows the parsing performance. Note that the overall tagging performance of the Berkeley parser is significantly worse than sequence models. However, better POS tagging does not lead to better parsing. The experiments suggest that sequence models propagate too many errors to the parser. Our linguistic analysis can also well explain the poor performance of Chinese CCG parsing when applying the C&C parser (Tse and Curran, 2012). We think the failure is mainly due to overplaying sequence models in both POS tagging and supertag-

<sup>1</sup>[www.csie.ntu.edu.tw/~cjlin/liblinear/](http://www.csie.ntu.edu.tw/~cjlin/liblinear/)

<sup>2</sup>[code.google.com/p/berkeleyparser/](http://code.google.com/p/berkeleyparser/)

	LLM	First order LGLM		Second order LGLM	
		SP	PA	SP	PA
Baseline	93.96%	94.30%	94.24%	94.49%	94.33%
+Word clustering	94.75%	94.90%	94.80%	95.05%	94.96%
<b>+Word clustering+HMMLA</b>	<b>95.12%</b>	<b>95.19%</b>	<b>95.18%</b>	<b>95.14%</b>	<b>95.22%</b>
+Word clustering+PCFGLA	95.42%	95.50%	95.40%	95.56%	95.44%
+Word clustering+DEP	95.28%	95.22%	95.26%	95.29%	95.25%
+ALL	95.56%	95.61%	95.60%	95.53%	95.53%

Table 3: Tagging accuracies of different stacking models on the development data.

ging.

Devel.	LP	LR	F1
Berkeley	80.44	80.31	81.36
1or LGLM	80.38	79.48	79.93↓
2or LGLM	80.98	79.93	80.45↓
HMMLA	80.65	79.62	80.13↓
1or LGLM(HMMLA)	81.55	80.80	81.17↓
1or LGLM(PCFGLA)	82.84	81.75	82.29↑
1or LGLM(DEP)	82.69	81.68	82.18↑

Table 4: Parsing accuracies on the development data. *1or* and *2or* respectively denote first order and second order. *LGLM(X)* denotes a stacking model with *X* as the level-0 processing. All stacking models incorporate word clusters to improve the tagging accuracy.

To distinguish the predictive abilities of generative and discriminative models, we report the precision of the prediction of unknown words (UNK). Discriminative learning can define arbitrary (even overlapping) features which play a central role in tagging English unknown words. The difference between generative and discriminative learning in Chinese POS tagging is not that much, mainly because most Chinese words are compactly composed by a very few Chinese characters that are usually morphemes. This language-specific property makes it relatively easy to smooth parameters of a generative model.

### 3 Improving Tagging Accuracy via Stacking

In this section, we study a simple way of integrating multiple heterogeneous models in order to exploit their complementary strength and thereby improve tagging accuracy beyond what is possible by either model in isolation. The method integrates the heterogeneous models by allowing the outputs of the HMMLA, PCFGLA and DEP to de-

fine features for the LLM/LGLM.

#### 3.1 Stacked Learning

*Stacked generalization* is a meta-learning algorithm that has been first proposed in (Wolpert, 1992) and (Breiman, 1996). Stacked learning has been applied as a system ensemble method in several NLP tasks, such as joint word segmentation and POS tagging (Sun, 2011), and dependency parsing (Nivre and McDonald, 2008). The idea is to include two “levels” of predictors. The first level includes one or more predictors  $g_1, \dots, g_K : \mathbb{R}^d \rightarrow \mathbb{R}$ ; each receives input  $\mathbf{x} \in \mathbb{R}^d$  and outputs a prediction  $g_k(\mathbf{x})$ . The second level consists of a single function  $h : \mathbb{R}^{d+K} \rightarrow \mathbb{R}$  that takes as input  $\langle \mathbf{x}, g_1(\mathbf{x}), \dots, g_K(\mathbf{x}) \rangle$  and outputs a final prediction  $\hat{y} = h(\mathbf{x}, g_1(\mathbf{x}), \dots, g_K(\mathbf{x}))$ . The predictor, then, combines an ensemble (the  $g_k$ ’s) with a meta-predictor ( $h$ ).

#### 3.2 Applying Stacking to POS Tagging

We use the LLMs or LGLMs (as  $h$ ) for the level-1 processing, and other models (as  $g_k$ ) for the level-0 processing. The characteristic of discriminative learning makes LLMs/LGLMs very easy to integrate the outputs of other models as new features. We are relying on the ability of discriminative learning to explore informative features, which play a central role in boosting the tagging accuracy. For output labels produced by each auxiliary model, five new *label uni/bi-gram* features are added:  $w_{-1}, w, w_{+1}, w_{-1-w}, w_{-w+1}$ . This choice is tuned on the development data.

Word clusters that are automatically acquired from large-scale unlabeled data have been shown to be very effective to bridge the gap between high and low frequency words, and therefore significantly improve PA tagging, as well as other syntactic processing tasks. Our stacking models are all built on word clustering enhanced discriminative linear models. Five *word cluster uni/bi-gram* features are

added:  $w_{-1}$ ,  $w$ ,  $w_{+1}$ ,  $w_{-1-w}$ ,  $w_{-w+1}$ . The clusters are acquired based on the Chinese giga-word data with the MKCLS tool. The number of total clusters is set to 500, which is tuned by (Sun and Uszkoreit, 2012).

### 3.3 Evaluation

Table 3 summarizes the tagging accuracy of different stacking models. From this table, we can clearly see that the new features derived from the outputs of other models lead to substantial improvements over the baseline LLM/LGLM. The output structures provided by the PCFGLA model are most effective in improving the LLM/LGLM baseline systems. Among different stacking models, the syntax-free hybrid one (i.e., stacking LLM/LGLM with HMMLA) does not need any treebank to train their systems. For the situations that parsers are not available, this is a good solution. Moreover, the decoding algorithms for linear-chain Markov models are very fast. Therefore the syntax-free hybrid system is more appealing for many NLP applications.

Table 5 is the F1 scores of the DEC/DEG prediction which are obtained by different stacking models. Compared to Table 1, we can see that the hybrid sequence model is still not good at handling long-distance ambiguities. As a result, it harms the parsing performance (see Table 4), though it achieves higher overall precision.

Devel.	DEC	DEG
1or LGLM(HMMLA)	82.93	86.64
1or LGLM(PCFGLA)	88.11	91.12
1or LGLM(DEP)	87.46	89.86

Table 5: F1 score of the *DEC/DEG* prediction of different stacking models on the development data.

### 3.4 Related Work

(Sun and Uszkoreit, 2012) introduced a Bagging model to effectively combine the outputs of individual systems. In the training phase, given a training set  $D$  of size  $n$ , the Bagging model generates  $m$  new training sets  $D_i$ 's by sampling examples from  $D$ . Each  $D_i$  is separately used to train  $k$  individual models. In the tagging phase, the  $km$  models outputs  $km$  tagging results, each word is assigned one POS label. The final tagging is the voting result of these  $km$  labels. Although this model is effective, it is too expensive in the sense

that it uses parser multiple times. We also implement their method and compare the results with our stacking model. We find the accuracy performance produced by the two different methods are comparable.

(Rush et al., 2010) introduced dual decomposition as a framework for deriving inference algorithms for serious combinatorial problems in NLP. They successfully applied dual decomposition to the combination of a lexicalized parsing model and a trigram POS tagger. Despite the effectiveness, their method iteratively parses a sentence many times to achieve convergence, and thus is not as efficient as stacking.

## 4 Improving Tagging Efficiency through Unlabeled Data

### 4.1 The Idea

Hybrid structured models often achieve excellent performance but can be slow at test time. In our problem, it is obviously too inefficient to improve POS tagging by parsing a sentence first. In this section, we explore unlabeled data to transfer the predictive power of hybrid models to sequence models. The main idea behind this is to use a fast model to approximate the function learned by a slower, larger, but better performing ensemble model. Unlike the true function that is unknown, the function learned by a high performing model is available and can be used to label large amounts of pseudo data. A fast and expressive model trained on large scale pseudo data will not overfit and will approximate the function learned by the high performing model well. This allows a slow, complex model such as massive ensemble to be compressed into a fast sequence model such as a first order LGLM with very little loss in performance.

This idea to use unlabeled data to transfer the predictive power of one model to another has been investigated in many areas, for example, from high accuracy neural networks to more interpretable decision trees (Craven, 1996), from high accuracy ensembles to faster and more compact neural networks (Bucila et al., 2006), or from structured prediction models to local classification models (Liang et al., 2008),

### 4.2 Reducing Hybrid Models to Sequence Models

For English POS tagging, Liang, Daumé and Klein (2008) have done some experiments to

Size of data	HMMLA	win size=3		win size=4		Voting	
		LLM	LGLM	LLM	LGLM	DEC/DEG	
+100k	94.72%	95.05%	95.07%	95.04%	95.10%	95.36%	--
+200k	94.77%	95.06%	95.18%	95.20%	95.23%	95.43%	--
+500k	94.97%	95.11%	95.21%	95.15%	95.23%	95.43%	--
+1000k	95.09%	95.19%	95.23%	95.22%	95.31%	95.49%	85.75/89.01

Table 6: Tagging accuracies of different re-compiled models on the development data.

transfer the power of a chain conditional random field to a logistic regression model. Similarly, we do some experiments to explore the feasibility of reducing hybrid tagging models to a HMMLA, LLM or LGLM, for Chinese POS tagging. The large-scale unlabeled data we use in our experiments comes from the Chinese Gigaword (LDC2005T14), which is a comprehensive archive of newswire text data that has been acquired over several years by the Linguistic Data Consortium (LDC). We choose the Mandarin news text, i.e. Xinhua newswire. We tag giga-word sentences by applying the stacked first order LGLMs with all other models. In other words, the HMMLA, PCFGLA and DEP systems are applied to tag unlabeled data features and their outputs are utilized to define features for first-order and second-order LGLMs which produce pseudo training data. Both original gold standard training data and pseudo training data are used to re-train a HMMLA, a LLM/LGLM with extended features.

The key for the success of hybrid tagging models is the existence of a large diversity among learners. Zhou (2009) argued that when there are lots of labeled training examples, unlabeled instances are still helpful for hybrid models since they can help to increase the diversity among the base learners. The author also briefly introduced a preliminary theoretical study. In this paper, we also combine the re-trained models to see if we can benefit more. We utilize voting as the strategy for final combination. In the tagging phase, the re-trained LLM, LGLM and HMMLA systems outputs 3 tagging results, each word is assigned one POS label. The final tagging is the voting result of these 3 labels.

## 4.3 Experiments

### 4.3.1 Reducing Hybrid Models to HMMLA

With the increase of (pseudo) training data, a HMMLA may learn better latent variables to sub-categorize POS tags, which could significantly im-

prove a purely supervised HMMLA. In our experiments, all HMMLA models are trained with 8 iterations of split, merge, smooth. The second column of Table 6 shows the performance of the re-trained HMMLAs. The first column is the number of sentences of pseudo sentences. The pseudo sentences are selected from the beginning of the Chinese gigaword. We can clearly see that the idea to leverage unlabeled data to transfer the predictive ability of the hybrid model works. Self-training can also slightly improve a HMMLA (Huang et al., 2009). Our auxiliary experiments show that self-training is not as effective as our methods.

### 4.3.2 Reducing Hybrid Models to LLM/LGLM

To increase the expressive power of a discriminative classification model, we extend the feature templates. This strategy is proposed by (Liang et al., 2008). In our experiments, we increase the window size of word uni/bi-gram features to approximate long distance dependencies. For window size 3, we will add  $w_{-3}$ ,  $w_3$ ,  $w_{-3}w_{-2}$  and  $w_2w_3$  as new features; for size 4, we will add  $w_{-4}$ ,  $w_{-3}$ ,  $w_3$ ,  $w_4$ ,  $w_{-4}w_{-3}$ ,  $w_{-3}w_{-2}$ ,  $w_2w_3$  and  $w_3w_4$ ; Column 3 to 6 of Table 6 show the performance of the re-compiled LLMs/LGLMs. Similar to the generative model, the discriminative LLM/LGLM can be improved too.

### 4.3.3 Voting

The last two columns of Table 6 are the final voting results of the HMMLA, LLM and LGLM. The window size of word uni/bi-gram features for the LLM and LGLM is set to 4. Obviously, the re-trained models are still diverse and complementary, so the voting can further improve the sequence models. The result of the best hybrid sequence model is very close to the best stacking models. Furthermore, the F1 scores of the DEC/DEG prediction are 85.75 and 89.01, which are very close to parsers too.

#### 4.3.4 Improving Parsing

Purely supervised sequence models are not good at predicting function words, and accordingly are not good enough to be used as front modules to parsers. The re-compiled models can mimic some behaviors of parsers, and therefore are suitable for parsing. Our evaluation shows that the significant improvement of the POS tagging stop harming syntactic parsing. Results in Table 7 indicate that the parsing accuracy of the Berkeley parser can be simply improved by inputting the Berkeley parser with the re-trained sequential tagging results. Additionally, the success to separate tagging and parsing can improve the whole syntactic processing efficiency.

Devel.	LP	LR	F1
HMMLA	82.18	81.16	81.66↑
LLM	81.86	80.93	81.40↑
LGLM	82.07	81.21	81.64↑
Voting	82.34	81.42	81.88↑

Table 7: Accuracies of parsing based on re-compiled tagging.

#### 4.3.5 Final results

Table 8 shows the performance of different systems evaluated on the test data. Our final sequence model achieve the state-of-the-art performance, which is once obtained by combining multiple parsers as well as sequence models.

Systems	Acc.
(Sun and Uszkoreit, 2012)	95.34%
Our system	95.33%

Table 8: Tagging accuracies on the test data.

## 5 Conclusion

In this paper, we study two techniques to build accurate and fast sequence models for Chinese POS tagging. In particular, our goal is to capture long-distance dependencies in sequence models. To improve tagging accuracy, we study stacking to integrate multiple models with heterogeneous views. To improve tagging efficiency at test time, we explore unlabeled data to transfer the predictive power of hybrid models to simple sequence or even local classification models. Hybrid systems are utilized to create large-scale pseudo training data for cheap models. By applying complex

machine learning techniques, we are able to build good sequential POS taggers. Another advantage of our system is that it serves as a front-end to a parser very well. Our study suggests that complicated structured models can be well simulated by simple sequence models through unlabeled data.

## Acknowledgement

The work was supported by NSFC (61170166), Beijing Nova Program (2008B03) and National High-Tech R&D Program (2012AA011101).

## References

- Leo Breiman. 1996. Stacked regressions. *Machine Learning*, 24:49–64, July.
- Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*, pages 535–541.
- Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL*.
- Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage ccg parsing. In *Proceedings of Coling 2004*, pages 282–288, Geneva, Switzerland, Aug 23–Aug 27. COLING.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8. Association for Computational Linguistics, July.
- Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.
- 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.
- Mark Craven. 1996. *Extracting Comprehensible Models from Trained Neural Networks*. Ph.D. thesis, University of Wisconsin-Madison, Department of Computer Sciences. Also appears as UW Technical Report CS-TR-96-1326.
- Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009)*, June 4-5, Boulder, Colorado, USA.



- Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint POS tagging and dependency parsing in chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1216–1224, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.
- Zhongqiang Huang, Vladimir Eidelman, and Mary Harper. 2009. Improving a simple bigram hmm part-of-speech tagger by latent annotation and self-training. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 213–216, Boulder, Colorado, June. Association for Computational Linguistics.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for Chinese POS tagging and dependency parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1180–1191, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Percy Liang, Hal Daumé, III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proceedings of the 25th international conference on Machine learning, ICML '08*, pages 592–599, New York, NY, USA. ACM.
- Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of ACL, ACL '05*, pages 75–82, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, Ohio, June. Association for Computational Linguistics.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.
- Alexander M Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of EMNLP*, pages 1–11, Cambridge, MA, October. Association for Computational Linguistics.
- Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. 2012. Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of ACL*, pages 440–448, Jeju Island, Korea, July. Association for Computational Linguistics.
- Weiwei Sun and Hans Uszkoreit. 2012. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate Chinese part-of-speech tagging. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, July.
- Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1385–1394, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Daniel Tse and James R. Curran. 2012. The challenges of parsing chinese with combinatory categorial grammar. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 295–304, Montréal, Canada, June. Association for Computational Linguistics.
- Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Fast full parsing by linear-chain conditional random fields. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 790–798, Athens, Greece, March. Association for Computational Linguistics.
- David H. Wolpert. 1992. Original contribution: Stacked generalization. *Neural Netw.*, 5:241–259, February.
- Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.
- Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the Chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 162–171, Paris, France, October. Association for Computational Linguistics.
- Zhi-Hua Zhou. 2009. When semi-supervised learning meets ensemble learning. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems, MCS '09*, pages 529–538, Berlin, Heidelberg. Springer-Verlag.

# Exploring Semantic Information in Hindi WordNet for Hindi Dependency Parsing

Sambhav Jain Naman Jain Aniruddha Tammewar  
Riyaz Ahmad Bhat Dipti Misra Sharma

Language Technologies Research Centre  
IIIT Hyderabad

{sambhav.jain, riyaz.bhat}@research.iiit.ac.in, dipti@iiit.ac.in  
{naman.jain, uttam.tammewar}@students.iiit.ac.in

## Abstract

In this paper, we present our efforts towards incorporating external knowledge from Hindi WordNet to aid dependency parsing. We conduct parsing experiments on Hindi, an Indo-Aryan language, utilizing the information from concept ontologies available in Hindi WordNet to complement the morpho-syntactic information already available. The work is driven by the insight that concept ontologies capture a specific real world aspect of lexical items, which is quite distinct and unlikely to be deduced from morpho-syntactic information such as morph, POS-tag and chunk. This complementing information is encoded as an additional feature for data driven parsing and experiments are conducted. We perform experiments over datasets of different sizes. We achieve an improvement of 1.1% (LAS) when training on 1,000 sentences and 0.2% (LAS) on 13,371 sentences over the baseline. The improvements are statistically significant at  $p < 0.01$ . The higher improvements on 1,000 sentences suggest that the semantic information could address the data sparsity problem.

## 1 Introduction

Last decade has witnessed several efforts towards developing robust data driven dependency parsing techniques (Kübler et al., 2009). The efforts, in turn, initiated a parallel drive for building dependency annotated treebanks (Tsarfaty et al., 2013), which serve as a data source for training data driven dependency parsers. The annotations are often multi-layered and furnish information on part of speech category of word forms, their morphological features, related word groups and the

syntactic relations. The availability of such rich resources have considerably improved the parsing performance of syntactic parsers (Collins et al., 1999). However, the error analysis studies carried out on these parsers later revealed that certain syntactic relations are difficult to deduce and disambiguate with the syntactic information available in the annotated treebanks.

The need for richer information invoked several efforts in the direction of annotating higher order linguistic information in treebanks. It was felt that semantics can be leveraged for syntactic disambiguation and thus semantic annotation was performed in syntactic treebanks to complement the morpho-syntactic annotations (Kingsbury et al., 2002; Montemagni et al., 2003). Fujita et al. (2007) and MacKinlay et al. (2012) illustrated that semantic annotation delivers a significant improvement in parsing, confirming the hypothesis that semantics can assist syntactic analysis.

Among Indian languages, notable efforts on using semantic information in dependency parsing are on *Hindi*. Bharati et al. (2008) illustrated that mere *animacy* (human, non-human and inanimate) of a nominal significantly improves the accuracy of the parser. Later studies on extending such information with finer semantic distinctions like *time*, *place*, *abstract* reconfirmed the substantial role of semantics in syntactic parsing (Ambati et al., 2009). These studies are carried out on a dataset with hand annotated semantics. Although these studies provide deep insights on the role of semantics in parsing, they are limited in application as such information can not be automatically generated while parsing new sentences.

In this work, we make an effort to supply the aforementioned semantic information by employing concept hierarchy available in Hindi WordNet (henceforth HWN).

## 2 Related Work

Attempts have been made to utilize hand annotated semantic information for *constituency parsing* (Fujita et al., 2007; MacKinlay et al., 2012) as well as *dependency parsing* (Øvrelid and Nivre, 2007; Bharati et al., 2008; Ambati et al., 2009). However, acquiring such information for new sentences remains a challenge. This leads us to the exploration of lexical databases and ontologies for accessing semantic information useful for parsing. Xiong et al. (2005) used two lexical resources *HowNet*<sup>1</sup> (Dong and Dong, 2000) and *TongYiCi CiLin* (Mei and Gao, 1996) for parsing Penn Chinese Treebank (Xue et al., 2002). Agirre et al. (2008) demonstrated that semantic classes obtained from English WordNet (Miller, 1995) help to obtain significant improvements in both PP attachment and PCFG parsing. Similarly, for dependency parsing, Agirre et al. (2011) utilized the English WordNet semantic classes and improved parsing accuracies.

## 3 Background and Challenges

Hindi is an Indo-Aryan language with richer morphology as compared to English. It exerts a relatively free word order with SOV being the default configuration. Due to the flexible word order, dependency representations are preferred over constituency for its syntactic analysis (Bharati and Sangal, 1993). The dependency representations do not constrain the order of words in a sentence and thus are better suited for flexible ordering of words. The dependency grammar formalism, used for Hindi is *Computational Paninian Framework* (CPG) (Begum et al., 2008; Bharati et al., 2009). The dependency relations in CPG formalism are closer to semantics and hence they are also denoted as *syntactico-semantic* relations.

The most important feature explored for dependency parsing is ‘case clitics’ that largely governs the relations nominals bear with their heads. Several efforts in past, on parsing Hindi, have greatly benefited by utilizing these clitics as a feature (Ambati et al., 2010a; Ambati et al., 2010b). However, case markers and case roles do not have a one-to-one mapping, each case marker is distributed over a number of case roles. Among the six case markers only Ergative case marker is unambiguous (Mohanan, 1994). Although case

markers are good indicators of the relation a nominal bears in a sentence, their ambiguous nature bars their ability in effectively identifying the role of a nominal while parsing. Consider the examples from (1a-e), the instrumental *se* is extremely ambiguous. It can mark the instrumental adjuncts as in (1a), source expressions as in (1b), material as in (1c), comitatives as in (1d), and causes as in (1e).

- (1a) मोहन ने चाबी से ताला खोला ।  
Mohan-Erg key-Inst lock-Nom open  
‘Mohan opened the lock with a key.’
- (1b) गीता ने दिल्ली से सामान मंगवाया ।  
Geeta-Erg Delhi-Inst luggage-Nom procure  
‘Geeta procured the luggage from Delhi.’
- (1c) मूर्तिकार ने पत्थर से मूर्ति बनायी ।  
sculptor-Erg stone-Inst idol-Nom make  
‘The sculptor made an idol out of stone.’
- (1d) राम की श्याम से बात हुई ।  
Ram-Gen Shyaam-Inst talk-Nom happen  
‘Ram spoke to Shyaam.’
- (1e) बारिश से कई फसलें तबाह हो गयीं ।  
rain-Inst many crops-Nom destroy  
happen-Perf  
‘Many crops were destroyed due to the rain.’

Not all instances of a nominal in Hindi are case marked, as shown in Table 1. In appropriate contexts, a nominal can also bear a nominative case which is morphologically null (henceforth referred as unmarked nominals). It is possible, in fact quite frequent, to have more than one unmarked nominal within a single clause and due to the relative free word order, the movement can result in different surface configurations.

- (2a) चिड़िया दाना चुग रही है ।  
bird-Nom grain-Nom peck-Prog
- (2b) दाना चिड़िया चुग रही है ।  
grain-Nom bird-Nom peck-Prog  
‘A bird is pecking grain.’

	Patient-Unmarked	Patient-Marked
Agent-Unmarked	1276	741
Agent-Marked	5373	966

Table 1: Co-occurrence of Marked and Unmarked verb arguments in Hindi Dependency Treebank. *Source:* training-set, shared task MTPIL 2012

A conventional parser has no cues for the disambiguation of instrumental case marker *se* in examples (1a-e) and similarly, in example (2a-b), it

<sup>1</sup><http://www.keenage.com>

is hard for the parser to know whether ‘bird’ or ‘grain’ is the agent of the action ‘peck’. Apart from lexical and structural ambiguity, there are also data sparsity and out of vocabulary (OOV) problems when parsing out-of-domain text. Traditionally, syntactic parsing has largely been limited to the use of only a few lexical features. Features like POS-tags are way too coarse to provide deep information valuable for syntactic parsing. So in order to assist the parser for better judgments, we need to complement the morphology somehow.

#### 4 Hindi WordNet and Concept Ontologies

Hindi WordNet is a lexical database developed on the lines of English Wordnet, under the Indo WordNet project (Narayan et al., 2002). For each lexical item, Hindi WordNet defines a synset which enlists its synonyms. Further, each synset is mapped to a concept ontology. The concept ontology is a hierarchical organization of concepts like entities, actions etc. which defines the semantic properties of lexical items of a given synset. The ontology consists of around 200 different concepts. The lexical item is the leaf node in this hierarchical construct. As we move up the hierarchy, the specific semantic aspects of a given lexical item are unraveled. The hierarchy terminates, immediately after capturing the syntactic category of a word, at the *TOP* node. The *TOP* acts as a *root*, holding the hierarchies of all the lexical items listed in HWN. Figure 1 illustrates a typical hierarchy in this ontology, where *Ape* is the most explanatory node. As we move up, it becomes more and more generic. Further, the relations between different synsets are captured based on the following paradigms :

- Semantic (hypernymy, hyponymy, meronymy etc.)
- Lexical (antonymy, synonymy etc.)
- Gradience (size, quality, manner etc.).

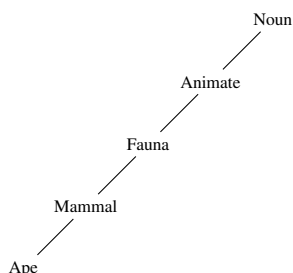


Figure 1: Sample Hierarchy of Concepts in Hindi Wordnet

Type	Sentence Count	Token Count	Chunk <sup>4</sup> Count
Training	12,038	268,009	142,445
Development	1,233	26,416	13,945
Testing	1,828	39,775	21,165

Table 2: Statistics of Data Sets used for experiments

#### 5 Hindi Dependency Treebank

In this section, we give an overview of Hindi Treebank (HTB ver-0.51) (Bhatt et al., 2009; Palmer et al., 2009) a part of which was released for Hindi Dependency Parsing shared task, MTPIL, (Sharma et al., 2012). It is a multi-layered dependency treebank with morphological, part-of-speech and dependency annotations based on the Computational Paninian Framework (henceforth CPG). In the dependency annotation, relations are mainly verb-centric. The relation that holds between a verb and its arguments is called a ‘*karaka*’ relation. Besides *karaka* relations, dependency relations also exist between nouns (genitives), between nouns and their modifiers (adjectival modification, relativization), between verbs and their modifiers (adverbial modification including subordination). CPG provides an essentially syntactico-semantic dependency annotation, incorporating *karaka* (e.g., agent, theme, etc.), *non-karaka* (e.g. possession, purpose) and other (part of) relations. A complete tag-set of dependency relations based on CPG can be found in (Bharati et al., 2009). The ones starting with ‘*k*’ are largely Paninian *karaka* relations, and are assigned to the arguments of a verb. The data is released in two formats, SSF (Bharati et al., 2007) and CoNLL-X<sup>2</sup> formats (details in Table 2). It has also been released in UTF-8 encoding and roman readable WX<sup>3</sup> notation. We are using the CoNLL-X format and UTF-8 encoding.

#### 6 Incorporating Knowledge from Concept Ontologies

In this section, we present our approach to incorporate semantic knowledge from HWN into the parsing model. We transform the hierarchical information in the concept ontology listed in HWN, into a string feature (henceforth WN feature) for

<sup>2</sup><http://ilk.uvt.nl/conll/#dataformat>

<sup>3</sup><http://sanskrit.inria.fr/DATA/wx.html>

<sup>4</sup>A chunk is a set of adjacent words which are in dependency relation with each other, and are connected to the rest of the words by a single incoming arc.

all the tokens in our data. Given a lexical item, we extract the information using its syntactic category from the ontological hierarchy corresponding to the most appropriate sense selected. In the following, we discuss in detail the selection and incorporation of this information with the challenges posed.

## 6.1 Feature Extraction

In this section, we explore the extraction of features from HWN corresponding to the lexical items in our data. We also address the issues like sense selection and coverage.

### 6.1.1 Sense Selection

Attributed to the phenomenon of lexical ambiguity, a lexical item can have senses varying across different contexts. Although HWN lists all the possible senses of a lexical item, to choose the contextually appropriate sense is a challenging task. Here, we discuss our approach to select the sense of a lexical item best suited in a given context.

- *Category Based Sense Selection*: Consider a word *chaat*, it can either mean ‘lick’ or ‘snacks’. The former corresponds to a verb while the latter is a nominal as depicted in Figure 2. The syntactic category of a lexical item provides an initial cue for the sense selection. Among the varied senses, we filter out the senses that do not fall into its syntactic category.

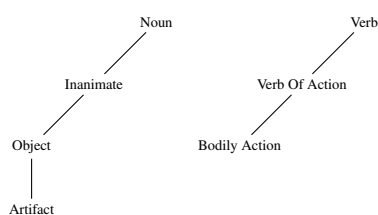


Figure 2: Nominal and Verb Sense of *chaat*

- *Intra – Category Sense Selection*: As a matter of fact, words are ambiguous not only across different syntactic categories but also within same category as depicted in Figure 3. Once the senses of a lexical item are filtered based on its syntactic category, within category senses, if many, are investigated for the best sense based on the following strategies:

- *First Sense*: Among the varied senses, we select the first sense listed

in HWN corresponding to the POS-tag of a given lexical item. The choice is motivated by our observation that the senses of a lexical item are ordered in the descending order of their frequencies of usage i.e., the first sense listed in HWN is the predominant sense of a given lexical item.

- *WSD*: Although first sense captures the predominant usage of a lexical item, it is inappropriate for its other infrequent usages. We, therefore, need to pick the contextually appropriate sense of a lexical item. To this end, we exercise Extended Lesk, a classical word sense disambiguation algorithm (Banerjee and Pedersen, 2003).

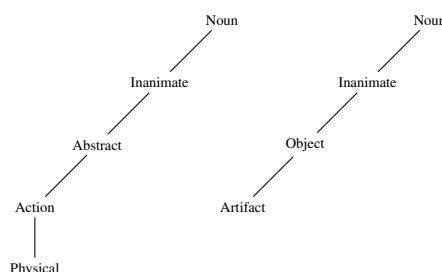


Figure 3: Two senses for the nominal *chaat*

### 6.1.2 Numeric Expressions

As is obvious, no lexical resource can have an exhaustive coverage because of the evolving nature of human language. In the context of HWN, the problem further intensifies as it restricts the entry to only words of open class syntactic categories. Apart from that, it also has a limited coverage for numeric expressions as these expressions belong to an infinite set. Numerals can be used in wide range of senses. Apart from their simple ordinal or cardinal usages, they can also be used as nominals in expressions like time and measurement. In their adjectival sense, WN features can be extracted corresponding to the head word they modify e.g., the temporal sense of an expression 10 *saal* can be identified by the head word *saal* ‘year’. However, to identify the temporal sense of a numeral, used as nominal, like 2013 is challenging. We use a numeric-expression recognizer, built in-house, to identify measurement and temporal expressions. The tool makes use of regular expressions and cue words. Once identified, we assign them an appropriate HWN ontological hierarchy which either corresponds to *time*, *measurement* or *number*.

### 6.1.3 Complex Predicate as a Feature

Complex predicates (CPs, also known as complex verbs) are highly frequent in South Asian languages (Mohan, 1997). They occur in the form of nominal+verb combinations (called conjunct verbs) and verb+verb combinations (called compound verbs). For example, in (5), ‘शरण लेना’ (refuge take) is a complex predicate composed of a nominal ‘शरण’ and a light verb ‘लेना’. The constituents of a complex predicate are related by a dependency relation *poF* in HDT. In Hindi dependency parsing, the major chunk of parse errors is attributed to the low learnability of complex predicates (Husain and Agrawal, 2012). Begum et al. (2011) addressed the identification of these expressions using some linguistic rules. Fortunately, HWN has listed a finite set of these expressions in its database (Chakrabarti et al., 2007). We first extract the multi word expressions listed in HWN if the last word in the expression is a verb. Then from the list only 2-word expressions are selected and treated as complex predicates. Instead of adding WN features to the nominal of a complex predicate, we assign a separate *CP* tag to it. The semantics of light verbs is, however, kept as such.

## 6.2 Feature Design

After the extraction of WN features, we explore possibilities of their design and incorporation in the parsing framework, as follows.

### 6.2.1 Grouping Similar Features

We observed that few concept ontological lineages are semantically similar. For example, the six lineages depicted below address the notion of time.

- *Time*
- *Descriptive*→*Time*
- *Inanimate*→*Abstract*→*Time*
- *Inanimate*→*Abstract*→*Time*→*Period*
- *Inanimate*→*Abstract*→*Time*→*Season*
- *Inanimate*→*Abstract*→*Time*→*Mythological Period*

Since our focus is on adding representative semantic features which can assist parsing, we believe that such divergences should be grouped together. In the listed example, first, second and the last four differ in terms of their origin and belong to different branches in the hierarchy. Thus they can not be grouped by optimal depth selection (described later in Section 6.2.3) and requires a manual scrutiny. We studied the possible lineages in the concept ontology and performed

merging wherever necessary, furnishing a semantically well diverse set of concept lineages.

### 6.2.2 Split Vs Conjoined

The concept lineage, derived for a word from HWN concept ontology, contains diverse concepts at each level of the lineage. The choice of using each of these concepts as independent features or the complete lineage as a single feature demands exploration. In the context of parsing, each independent concept from the lineage can potentially capture a specific aspect of syntax, depending on the fineness of the concept. The down side of this proposition is the increase in the feature dimensions, as each level adds a new dimension in the feature space. Whereas, using the complete lineage as a single feature does not add any additional dimension in the feature space but captures only a specific concept. This trade off is difficult to comprehend on theoretical grounds, hence we explore both choices of feature design in our experiments.

### 6.2.3 Ontology Depth

Hindi WordNet concept ontology furnishes a ‘generalization hierarchy’ for a lexical item, where the specificity of concepts increases as we move down the hierarchy. It may look intuitive to use fully expanded concept lineage, as it contains more detailed description of the lexical unit. However, opting for a highly fine-grained concept lineage leads to the problem of sparseness. It becomes less and less probable to find ample training examples as the feature becomes more fine-grained. At the same time, too much generalization is also unrewarding since the richer information is cast away in the excessive coarser lineage. This calls for measures to obtain an optimal depth of concept lineage for each lexical item. On one hand it should be generalized enough to give significant examples of its respective type while on the other hand, it should be fine enough to capture the rich ontological concept associated with the lexical unit. In order to quantify the trade-off we resort to statistical correlation measures and employed *Gini Coefficient* (Gini, 1912). We computed the coefficient against all possible concept lineages in the training set and set a threshold. The lineages that fall below the threshold are generalized till they are above the threshold. For example, in Figure 1 the concept *ape* is suppressed to give the lineage till *mammal* only. So in future if a word gives the lineage as in Figure 1 it will be

replaced with its one level up generalization i.e. *Animate*→*Fauna*→*Mammal*.

## 7 Experiments and Results

In our experiments, we focus on establishing dependency relations between the chunk heads which we henceforth denote as *inter-chunk* parsing. The relations between the tokens of a chunk (*intra-chunk* dependencies) are not considered for experimentation. In example (3), dotted line shows an *intra-chunk* relation while the bold lines show *inter-chunk* dependency relations<sup>5</sup>. The decision is motivated by the fact that the *intra-chunk* dependencies can easily be predicated automatically using a finite set of rules (Kosaraju et al., 2012). Moreover we also observed the high learnability of *intra-chunk* relations from an initial experiment. We found the accuracies of *intra-chunk* dependencies to be more than 99.00% for both Labeled Attachment and Unlabeled Attachment.

In this section, we present our parsing experiments incorporating the features extracted from HWN, as discussed in Section 6. First we setup our baseline parser followed by the detailed discussion on the impact of the individual features, extracted from HWN, on the overall parsing performance.

We setup our baseline parser on the lines of (Singla et al., 2012) with minor modifications in the parser *feature model*. We employ MaltParser version-1.7<sup>6</sup> (Nivre et al., 2007) and Nivre’s Arc Eager algorithm for all our experiments reported in this work. All the results reported are evaluated using *eval07.pl*<sup>7</sup>. We use MTPIL (Sharma et al., 2012) dependency parsing shared task data described in Section 5. Among the features available in the FEATS column of the CoNLL format data, we only consider *Tense*, *Aspect*, *Modality (tam)* and *postpositions* while training the baseline parser. Other columns like POS, LEMMA, etc. are used as such. After the baseline, the parsing framework is further enriched with the semantic features extracted from HWN to address the problems raised in Section 3. These features are added in the FEATS column of the data, separated by ‘|’. In a pilot experiment split form of features, as discussed in Section 6.2.2, are found to per-

<sup>5</sup>k1: Doer, k1s: Noun Complement, k5: Source, k7p: Place, k7t: Time, pof: part-of (complex predicate), lwg\_psp: local-word-group postposition

<sup>6</sup><http://www.maltparser.org/download.html>

<sup>7</sup><http://nextens.uvt.nl/depparse-wiki/SoftwarePage/#eval07.pl>

form better than conjoined form, which motivate us to use WN feature in split form in all our experiments. The experimentation proceeds in the order as listed in Table 3 which also presents the consolidated results of our parsing experiments using the MTPIL training and testing sets. In order to see the impact of semantic information on data sparsity, we split the MTPIL training set into datasets of different sizes. We experiment with 6 data sets of different sizes. The results are produced on MTPIL test set and are plotted on Graph (Figure 4). The increase in LS and LAS, as the training size decreases, shows the impact of semantic information on data sparsity. The improvement of 1.1 (LAS) by semantics upon reducing the training examples to 1000 implies that semantics can address the data sparsity and OOV problems when working with out-of-domain text.

Next we discuss the impact of WN features on the accuracy of our parsing results produced on datasets of different sizes:

- *Sense Selection*: As discussed in Section 6.1.1, we perform two experiments to extract the WN features corresponding to the most appropriate sense of a lexical item. In the first experiment, the first sense of each lexical item is selected while in the second, WSD is used to pick the contextually most appropriate sense. These features corresponding to the chosen sense are coupled with the features already present in the baseline. As depicted in Graph (Figure 4), there is a average increase of 0.38 (LAS) on all datasets using the first sense strategy from the baseline. However, using WSD the accuracy decreased across all datasets. As is obvious, the fall in accuracy can be attributed to the wrong sense selection. The problem can be addressed by using better WSD algorithms for Hindi.
- *Numeric Expressions and Grouping*: As discussed in Section 6.1.2, numeric expressions and sense grouping increases the coverage of HWN. This obvious reason is clearly depicted in the improvement in parsing results as shown in Table 3. More the semantic information available in the data, more will be its impact on the parsing.
- *Depth of Information*: The optimality of feature coarseness is put to test in this ex-

periment. This experiment is run on numeric expression data with feature pruning done as described in Section 6.2.3. An increment of average 0.03% LAS across datasets is observed from the previous experiment. In the test set, there are only a few cases that are updated by choosing an optimal lineage depth which explains the minimal increase in accuracy.

- *Complex Predicate*: As pointed in (Begum et al., 2011), addressing the low learnability of complex predicates can improve the parsing results. The improvements are particularly seen in the core arguments of a verb. The similar syntactic distribution of adjectival or nominal element of a complex predicate and the syntactic arguments of a verb particularly *objects*, make these expressions highly ambiguous. Identifying these expressions beforehand, as suggested in (Begum et al., 2011), improves the parsing performance. The incorporation of this crucial information from HWN is rewarding as we achieve an improvement of  $\sim 0.4\%$  in LAS on a dataset of 1,000 sentences.

	Experiments	LAS(%)	UAS(%)	LS(%)
E1	Baseline	83.69	92.43	86.58
E2	E1 + First Sense	83.78	92.4	86.73
E3	E1 + WSD (Extended Lesk)	83.6	92.34	86.57
E4	E2 + Numeric Expressions & Grouping	<b>83.88</b>	<b>92.45</b>	<b>86.87</b>
E5	E4 + Ontological Depth	83.84	92.4	86.79
E6	E4 + Complex Predicate	83.75	92.39	86.72
E7	E5 + E6 (Complex Predicate + Ontological Depth)	83.74	92.39	85.7

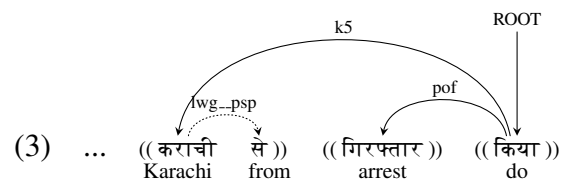
Table 3: Results of Parsing Experiments

## 8 Discussion

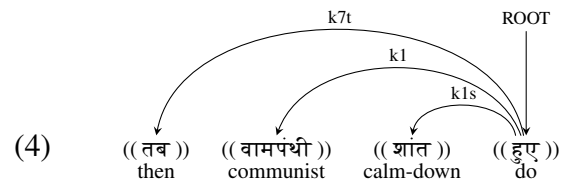
In this section, we discuss further, how well the issues raised in Section 3 are handled by the incorporation of semantic information in the parsing framework of Hindi. In Section 3, we stated that ambiguities in morphological cases in Hindi bar their efficient exploitation while parsing. Also we noted that unmarked nominals may as well affect the performance of a parser. So we propose semantics as a complementing information that can fill these gaps. Below we discuss whether semantic information has bridged these gaps or not.

- *Case Ambiguity*: Including the semantics from HWN to help disambiguate the con-

fusion present in a case marker, has improved parsing accuracy. Particularly confusion among the roles of concrete vs abstract time and place, and direct vs indirect object relations has been removed. In example (3), the dependency relation between nodes *Karachi* and *do* has been corrected from *k2* ‘Theme’ to *k5* ‘Source’. The post-position *from* can either mark a theme or a source relation. Semantics has removed this confusion.



- *Lack of Case Marker*: In absence of case marking lexical semantics acted as a complementing information. The improvement has been, as observed during error analysis, particularly for agents and patients. Thus semantics can be seen here as pseudo case markers. This is clearly visible from the example (4). The dependency relation between the nodes *then* and *do* has been corrected to *k7t* ‘time of action’ from *k1* ‘subject’.



- *Complex Predicates*: As we discussed, complex predicates are identified using HWN, so that the similar syntactic distributions of verb arguments and the nominal or adjectival part of a CP can be disambiguated. Identifying the complex predicates has turned to be rewarding. As was expected, the prior identification of CPs has significantly improved the joint identification of label and attachment. The system trained on 1,000 sentences has shown an improvement of 0.34% (LAS) and 0.2% (UAS) by prior identification of complex predicates. The confusion that has been removed is among the arguments of a verb and the nominal part of the CP i.e., between agent, patient vs nominal,



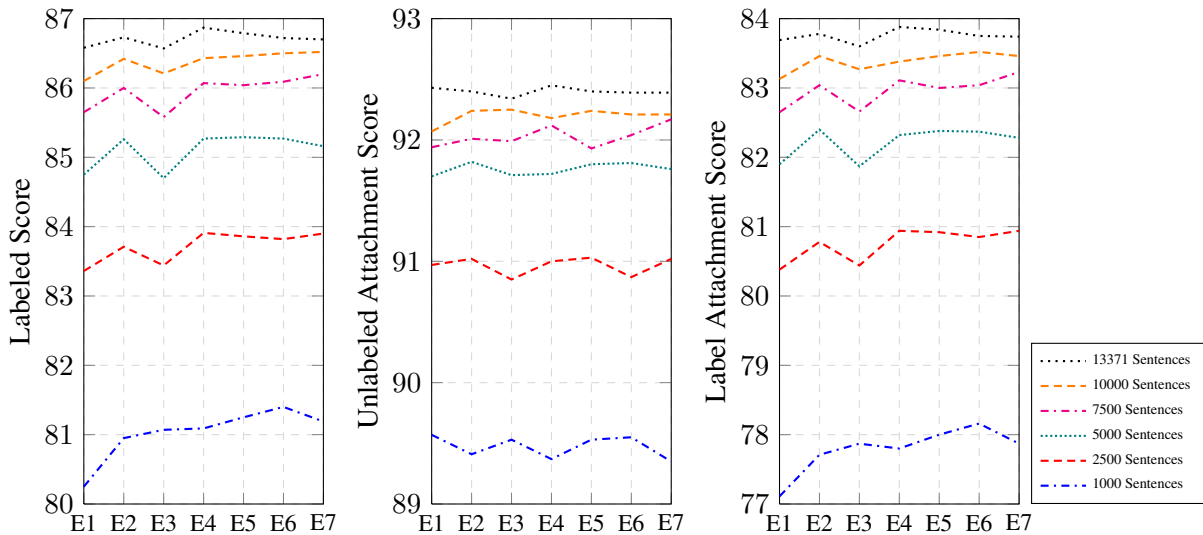
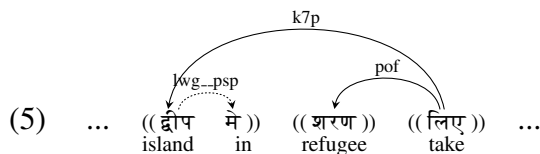


Figure 4: Impact of WN Features on Different Data Sizes

adjectival part of CP. In example below, baseline incorrectly identifies *refuge* as an argument of verb *take*. ‘*refuge take*’ is a complex predicate which is correctly identified upon incorporation of complex predicates in our parsing module.



## 9 Conclusion and Future Work

We present our efforts on exploring lexical resources, Hindi WordNet in our case, to discover features which complement the available morphosyntactic feature conventionally explored for parsing. We find concept ontology available in HWN quite resourceful in furnishing features which can essentially break syntactic ambiguity, resulting in better accuracies for parsing. In future we would like to investigate other hierarchies like hypernymy, hyponymy, meronymy etc. We would also like to substitute lexical units with their respective synsets as proposed in (Agirre et al., 2011).

## References

- E. Agirre, T. Baldwin, and D. Martinez. 2008. Improving parsing and PP attachment performance with sense information. *Proceedings of ACL-08: HLT*, pages 317–325.
- E. Agirre, K. Bengoetxea, K. Gojenola, and J. Nivre. 2011. Improving dependency parsing with semantic classes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 699–703.
- Bharat Ram Ambati, Pujitha Gade, Chaitanya Gsk, and Samar Husain. 2009. Effect of minimal semantics on dependency parsing. In *Proceedings of the Student Research Workshop*, pages 1–5, Borovets, Bulgaria, September. Association for Computational Linguistics.
- Bharat Ram Ambati, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010a. Two methods to incorporate local morphosyntactic features in Hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 22–30. Association for Computational Linguistics.
- B.R. Ambati, S. Husain, J. Nivre, and R. Sangal. 2010b. On the role of morphosyntactic features in Hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 94–102. Association for Computational Linguistics.
- Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 805–810. LAWRENCE ERLBAUM ASSOCIATES LTD.
- R. Begum, S. Husain, A. Dhvaj, D.M. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. In *Proceedings of IJCNLP*. Citeseer.
- Rafiya Begum, Karan Jindal, Ashish Jain, Samar Husain, and Dipti Misra Sharma. 2011. Identification of conjunct verbs in hindi and its effect on parsing accuracy. In *Computational Linguistics and Intelligent Text Processing*, pages 29–40. Springer.
- A. Bharati and R. Sangal. 1993. Parsing free word order languages in the Paninian framework. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 105–111. Association for Computational Linguistics.
- A. Bharati, R. Sangal, and D.M. Sharma. 2007. Ssf: Shakti standard format guide. *Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India*, pages 1–25.

- A. Bharati, S. Husain, B. Ambati, S. Jain, D. Sharma, and R. Sangal. 2008. Two semantic features make all the difference in parsing accuracy. *Proc. of ICON*, 8.
- A. Bharati, D.M. Sharma, S. Husain, L. Bai, R. Begum, and R. Sangal. 2009. AnnCorra: TreeBanks for Indian Languages Guidelines for Annotating Hindi Tree-Bank (version-2.0).
- R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D.M. Sharma, and F. Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189. Association for Computational Linguistics.
- Debasri Chakrabarti, Vijayanthi Sarma, and Pushpak Bhattacharyya. 2007. Complex predicates in indian language wordnets. *Lexical Resources and Evaluation Journal*, 40(3-4).
- Michael Collins, Lance Ramshaw, Jan Hajič, and Christoph Tillmann. 1999. A statistical parser for czech. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 505–512. Association for Computational Linguistics.
- Zhendong Dong and Qiang Dong. 2000. Hownet chinese-english conceptual database. Technical report, Technical Report Online Software Database, Released at ACL. <http://www.keenage.com>.
- Sanae Fujita, Francis Bond, Stephan Oepen, and Takaaki Tanaka. 2007. Exploiting Semantic Information for HPSG Parse Selection. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.
- Corrado Gini. 1912. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche*. Tipogr. di P. Cuppini.
- Samar Husain and Bhasha Agrawal. 2012. Analyzing Parser Errors to improve parsing accuracy and to inform tree banking decisions. *Linguistic Issues in Language Technology*, 7(1).
- Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology Conference*, pages 252–256. Citeseer.
- Prudhvi Kosaraju, Samar Husain, Bharat Ram Ambati, Dipti Misra Sharma, and Rajeev Sangal. 2012. Intra-chunk dependency annotation: expanding Hindi inter-chunk annotated treebank. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 49–56. Association for Computational Linguistics.
- Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127.
- Andrew MacKinlay, Rebecca Dridan, Diana McCarthy, and Timothy Baldwin. 2012. The effects of semantic annotations on precision parse ranking. 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*, pages 228–236. Association for Computational Linguistics.
- Jia-ju Mei and Yunqi Gao. 1996. Tongyi cilin (a chinese thesaurus). *China: Shanghai Lexicographical Publishing House*.
- George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.
- Tara Mohanan. 1994. *Argument structure in Hindi*. Stanford Univ Center for the Study.
- Tara Mohanan. 1997. Multidimensionality of representation: Nv complex predicates in hindi. *Complex predicates*, pages 431–471.
- Simonetta Montemagni, Francesco Barsotti, Marco Battista, Nicoletta Calzolari, Ornella Corazzari, Alessandro Lenci, Antonio Zampolli, Francesca Fanciulli, Maria Masettani, Remo Raffaelli, et al. 2003. Building the Italian syntactic-semantic treebank. In *Treebanks*, pages 189–210. Springer.
- Dipak Narayan, Debasri Chakrabarty, Prabhakar Pande, and Pushpak Bhattacharyya. 2002. An experience in building the indo wordnet-a wordnet for hindi. In *First International Conference on Global WordNet, Mysore, India*.
- Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kubler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95.
- Lilja Øvrelid and Joakim Nivre. 2007. When word order and part-of-speech tags are not enough—Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 447–451.
- M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D.M. Sharma, and F. Xia. 2009. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.
- Dipti Misra Sharma, Prashanth Mannem, Joseph vanGenabith, Sobha Lalitha Devi, Radhika Mamidi, and Ranjani Parthasarathi, editors. 2012. *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*. The COLING 2012 Organizing Committee, Mumbai, India, December.
- Karan Singla, Aniruddha Tammewar, Naman Jain, and Sambhav Jain. 2012. Two-stage Approach for Hindi Dependency Parsing Using MaltParser. *Training*, 12041(268,093):22–27.
- Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing Morphologically Rich Languages: Introduction to the Special Issue. *Computational Linguistics*, 39(1):15–22.
- D. Xiong, S. Li, Q. Liu, S. Lin, and Y. Qian. 2005. Parsing the penn chinese treebank with semantic knowledge. *Natural Language Processing-IJCNLP 2005*, pages 70–81.
- Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–8. Association for Computational Linguistics.

# Towards Robust Cross-Domain Domain Adaptation for Part-of-Speech Tagging

**Tobias Schnabel**

CIS, University of Munich  
tbs49@cornell.edu

**Hinrich Schütze**

CIS, University of Munich  
inquiries@cislmu.org

## Abstract

We investigate the robustness of domain adaptation (DA) representations and methods across target domains using part-of-speech (POS) tagging as a case study. We find that there is no single representation and method that works equally well for all target domains. In particular, there are large differences between target domains that are more similar to the source domain and those that are less similar.

## 1 Introduction

Domain adaptation (DA) is the problem of adapting a statistical classifier that was trained on a source domain (SD) to a target domain (TD) for which no or little training data is available. We present a case study that investigates the robustness of DA across six different TDs for POS tagging. Most prior work on DA has either been on a single TD, on two or more tasks – which results in an experimental setup in which two variables change at the same time, task and TD – or has not systematically investigated how robust different features and different DA approaches are.

The two main information sources in POS tagging are *context* – which POS's are possible in a particular syntactic context – and *lexical bias* – the prior probability distribution of POS's for each word. We address DA for lexical bias in this paper, focusing on unknown words; they are most difficult to handle in DA because no direct information about their possible POS is available in the SD training set. Since typical TDs contain a high percentage of unknown words, a substantial gain in the overall performance can be achieved by improving tagging for these words.

We address a problem setting where – in addition to *labeled SD data* – a large amount of *unlabeled TD data* is available, but *no labeled TD*

*data*. This setting is often called unsupervised domain adaptation (cf. (Daumé III, 2007)).

We make three contributions in this paper. First, we systematically investigate the cross-TD robustness of different representations and methods. We show that there are some elements of DA setups used in the literature that are robust across TDs – e.g., the use of distributional information – but that many others are not, including dimensionality reduction and shape information.

Second, we present an analysis that shows that there are two important factors that influence cross-TD variation: (i) the magnitude of the difference in distributional properties between SD and TD – more similar TDs require other methods than less similar TDs and (ii) the evaluation measures used for performance. Since in unsupervised DA we optimize learning criteria on a SD that can be quite different from the TD, different TD evaluation measures can diverge more in DA than in standard supervised learning settings when comparing learning methods.

Our third contribution is that we show that if we succeed in selecting an appropriate DA method for a TD, then performance improves significantly. We establish baselines for unknown words for the five TDs of the SANCL 2012 shared task and present the best DA results for unknown words on the Penn BioTreebank. Our improvements on this data set (by 10% compared to published results) are largely due to a new DA technique we call training set filtering. We restrict the training set to long words whose distribution is more similar to unknown words than that of words in general.

The next section describes experimental data and setup and Section 3 experimental results. Section 4 presents analysis and discussion. Section 5 reviews related work. Section 6 concludes.

## 2 Experimental data and setup

**Data.** Our SD is the Penn Treebank (Marcus et

al., 1993) of Wall Street Journal (WSJ) text. Following Blitzer et al. (2006), we use sections 2-21 for training. We also use 100,000 WSJ sentences from 1988 as unlabeled data in training.

We evaluate on six different TDs. The first TD is the Penn BioTreebank data set distributed by Blitzer. It consists of development and test sets of 500 sentences each and an unlabeled set of 100,000 sentences of BIO text.

The remaining five TDs (newsgroups, weblogs, reviews, answers, emails) are from the SANCL shared task (Petrov and McDonald, 2012). We will use WEB to refer to these five TDs collectively. Each WEB TD has an unlabeled training set of 100,000 sentences and development and test sets of about 1000 labeled sentences each. WEB and BIO tag sets differ slightly; we use them as published without modifications to make our results directly comparable to the benchmarks.

We define the *target domain repository* (TD-R) for a TD as the union of development set and unlabeled data available for that TD. SD+TD-R is the union of the source data (labeled and unlabeled WSJ) and TD-R.

**Classification setup.** In contrast to most other work on POS DA, we adopt a simple approach of *word classification*. The objects to be classified are words and the classes are the POS's of the SD. The gold label of a word in training is the majority tag in the SD. A prediction for an unknown word is then made by computing its feature representation and applying the learned classifier.

We adopt word classification instead of the more common sequence labeling setup because word classification is much more efficient to train and allows us to run a large number of experiments efficiently. Our experiments demonstrate that word classification accuracies are comparable with or higher than sequence labeling in POS DA for unknown words (cf. Table 2).

We use LIBSVM (Chang and Lin, 2011) to train  $\binom{k}{2}$  one-vs-one classifiers on the training set, where  $k$  is the number of POS tags in the latter. The SVMs were trained with untuned default parameters; in particular,  $C = 1$ . For sequence classification, we use CRFSuite (Okazaki, 2007), a Conditional Random Field (CRF) toolkit. Apart from the word features described below, we use the base feature set of Huang and Yates (2009) for CRFs, including features for state, emission and transition probabilities. CRFs are trained until

convergence with a limit of 300 iterations.

**Features.** There are in principle two sources of information to predict the POS of an unknown word in an unsupervised setting: the word itself (sequence of letters, shape etc) and the context(s) in which it occurs. For syntactic categorization, the immediate left and right neighbors of a word are the most informative aspect of context. Based on this reasoning, we create a feature representation for each word that has three components: left context information, right context information and shape information. We will refer to left/right context information as *distributional information*. Let  $f$  be the function that maps a word  $w$  to its (full) feature vector. We then define  $f$  as follows:

$$f(w) = \begin{bmatrix} f_{\text{left}}(w) \\ f_{\text{right}}(w) \\ f_{\text{shape}}(w) \end{bmatrix}$$

Based on the intuition that each of the three sources of information is equally important, each of the three component vectors is normalized to unit length.

For both distributional and shape features, we have a choice of either using *all possible features* or *a subset consisting of the most frequent features*. We directly compare these two possibilities, using recommended values from the literature for the subset condition: the 250 most frequent features (indicator words) for distributional vectors (Schütze, 1995) and the 100 most frequent features (suffixes) for shape vectors (Müller et al., 2012). Each component vector has an additional binary feature that is set to 1 if the rest of the vector is zero, and 0 otherwise to avoid numerical issues with zero vectors.

**Distributional features.** The  $i^{\text{th}}$  entry  $x_i$  of  $f_{\text{left}}(w)$  is the number of times that the *indicator word*  $t_i$  occurs immediately to the left of  $w$ :

$$x_i = \text{freq}(\text{bigram}(t_i, w))$$

where  $t_i$  is the word with frequency rank  $i$  in the corpus.  $f_{\text{right}}(w)$  is defined analogously.

Many different ways of defining and transforming distributional features have been proposed in the literature. We systematically investigate the following variables: (i) weighting (ii) dimensionality reduction and (iii) selection of data that distributional vectors are based on.

We experiment with three different *weighting functions* that transform non-zero counts as follows. (i) tf:  $w_{\text{tf}}(x) = 1 + \log(x)$ , (ii) tf-idf:

$w_{\text{tf-idf}}(x) = (N/\log \text{df}_{t_i})(1 + \log(x))$  (where  $N$  is the total number of words and  $\text{df}_{t_i}$  the number of words that indicator word  $t_i$  is a non-zero feature of) and (iii) binary:  $w_{\text{bin}}(x) = 1$ .

Transformation operations like *dimensionality reduction* (Deerwester et al., 1990) can be effective in improving generalization in machine learning, in particular in nonstandard settings like DA where a labeled random sample of the TD is not available. We test singular value decomposition (SVD) here because it has been used in prior work on POS (Huang and Yates, 2009). We apply SVD to the matrix of all feature vectors and keep the dimensions corresponding to the  $d = 100$  largest singular values.

We compute distributional vectors either on target data only (i.e., on TD-R) or on the union of source and target data (i.e., SD+TD-R). We compare these two alternatives and show in our experiments that SD distributional information does not consistently improve performance.

**Shape features.** Suffixes are likely to be helpful because regular processes of inflectional and derivational morphology do not change in English when going from one domain to the next. Many POS taggers incorporate information from suffixes to build robust features (Miller et al., 2007). For a selected suffix  $s$ , we simply set the dimension corresponding to  $s$  in  $f_{\text{shape}}(w)$  to 1 if  $w$  ends in  $s$  and to 0 otherwise. We either select all suffixes or the top 100, depending on the experiment.

In addition to suffixes, we investigate two other representational variables related to shape: case and digits. For case, we compare keeping case information as is with converting all uppercase characters to lowercase characters. For digits, we compare keeping digits as is with converting all digits to the digit 0; e.g., \$1,643 is converted to \$0,000). We call these two transformations *case normalization* and *digit normalization*.

**Training set filtering.** The key challenge in DA is that the distributions of source and target are different. One simple trick we can apply to make the distributions more similar is to eliminate all short words from the training set. We call this (training set) *filtering*. The reason this is promising is that longer words are more likely to be examples of productive linguistic processes than short words – even if this is only a statistical tendency with many exceptions.

In future work, we would like to test other fil-

tering options that are based on similar principles, including filtering based on word frequency and open/closed tag classes. Filtering on word length is simple and we show below that it is able to improve accuracy by several percentage points on one TD.

### 3 Experimental results

We train  $\binom{k}{2}$  binary SVM classifiers on the feature representations we just defined. The training set consists of all words that occur in the WSJ training set (in condition SD+TD-R) or all words that occur in both the WSJ training set and TD-R (in condition TD-R). An unknown word is classified by building its feature vector, running the classifiers on it and then assigning it to the POS class returned by the LIBSVM one-vs-one setup.

We divide our experiments into two parts. In the *basic experiment*, we investigate four parameters of the model that are likely to interact with each other: dimensionality of shape vectors (ALL vs. 100 most frequent suffixes), dimensionality of distributional vectors (ALL vs. 250 most frequent indicator words), use of dimensionality reduction (SVD: yes or no) and weighting of distributional vectors (bin, tf, tf-idf).

In the *extended experiment*, we then investigate the effect of other parameters on the best performing model from the basic experiment: distributional vectors based on SD+TD-R vs TD-R, case normalization, digit normalization, completely omitting either shape or distributional information and training set filtering. For the basic experiment, these parameters are set to the following values: distributional vectors are computed on TD-R, case normalization is used, digit normalization is not used, and the training set is not filtered (i.e., all words are included in the training set).

**Basic experiment.** Table 1 gives the results of the basic experiment: the 24 possible combinations of number of shape features, number of distributional features, use of dimensionality reduction and weighting scheme. In each column, the best three accuracies are underlined and the best accuracy is doubly underlined; the results significantly different from the best result are marked with a dagger.<sup>1</sup>

The goal of the basic experiment is to exhaus-

<sup>1</sup> $p < .05$ , 2-sample test for equality of proportions with continuity correction. We use the same test and level for all significance results in this paper.

shape	dist	svd	wght	grp	rev	blog	ans'r	em'l	BIO	
1	100	250	n	bin	<u>56.88</u>	63.92	67.13 †	52.14	63.30	65.64 †
2				tf	56.50	65.67	70.33	52.47	<u>64.37</u>	63.14 †
3				tf-idf	<u>57.14</u>	<u>65.83</u>	70.23	51.86 †	64.14	64.94 †
4			y	bin	<u>52.52</u> †	54.68 †	62.74 †	47.81 †	60.08 †	70.29 †
5				tf	54.42	58.18 †	68.01 †	48.14 †	61.70 †	69.70 †
6				tf-idf	54.73	57.44 †	68.75 †	48.93 †	61.38 †	<u>70.95</u> †
7	ALL	n	bin	55.98	63.60	68.70 †	52.14	62.87	68.92 †	
8				tf	56.58	64.67	70.82	51.02 †	63.52	65.72 †
9				tf-idf	56.15	63.50	68.85 †	50.09 †	61.87 †	68.61 †
10			y	bin	52.05 †	52.82 †	60.67 †	41.95 †	59.82 †	68.57 †
11				tf	53.65 †	57.23 †	66.24 †	43.02 †	61.22 †	69.82 †
12				tf-idf	54.21 †	55.47 †	64.17 †	42.50 †	58.52 †	69.11 †
13	ALL	250	n	bin	56.02	65.04	70.77	54.05	<u>64.37</u>	68.45 †
14				tf	55.59	<u>66.05</u>	<u>72.45</u>	<u>55.03</u>	<u>64.43</u>	64.82 †
15				tf-idf	55.93	<u>65.99</u>	<u>72.10</u>	<u>54.98</u>	63.98	65.87 †
16			y	bin	52.48 †	56.16 †	65.50 †	43.48 †	59.79 †	70.64 †
17				tf	53.26 †	59.46 †	68.95 †	48.51 †	60.60 †	68.68 †
18				tf-idf	54.16 †	59.56 †	68.70 †	44.18 †	60.66 †	69.35 †
19	ALL	n	bin	56.06	63.55	68.85 †	54.38	59.85 †	66.22 †	
20				tf	<u>56.62</u>	64.61	<u>71.86</u>	54.28	61.05 †	65.64 †
21				tf-idf	56.15	63.07	69.74	52.65	59.95 †	65.25 †
22			y	bin	52.35 †	55.74 †	62.89 †	41.95 †	58.68 †	<u>71.07</u> †
23				tf	53.99 †	59.83 †	68.16 †	43.62 †	60.37 †	69.93 †
24				tf-idf	54.81	58.98 †	68.65 †	41.95 †	58.68 †	<u>74.39</u>

Table 1: Accuracy of unknown word classification in the basic experiment. The performance of the best (three best) parameter combinations per column are doubly (singly) underlined. A dagger indicates a result significantly worse than the column’s best result.

tively investigate combinations of the four parameters that we suspect to have the strongest interaction with each other and then find a parameter combination that is a good basis for testing the remaining parameters in the extended experiment. The guiding principle in this investigation is that when in doubt, we select the simpler or default setting for the extended experiment in order to make as few assumptions as possible.

For the number of shape features, ALL generally does better than 100. Five TDs have their best result for ALL: rev, blog, answer, email (line 14) and BIO (line 24). The exception is grp (best result on line 3). The reason seems to be that the newsgroups TD contains a larger number of unknown words with suffixes that do not support POS generalization well. E.g., the suffixes -ding, -eding, -eeding, -breeding of a newsgroup name like “alt.animals.horses.breeding” (mistagged as VBG, gold tag: NN) are misleading. Despite these problems, the best 100 result for newsgroups is not significantly better than the best ALL result (lines 3 vs. 20). This argues for using the setting ALL for the extended experiment.

For the number of distributional features, there is a similar tendency for the WEB TDs (grp, rev, blog, answer, email) to do slightly better for fewer

features (250) than ALL features. However, BIO clearly benefits from using the full dimensionality of the distributional feature space: all 250 results are statistically worse than the best ALL result and the gap to the best 250 result is large (line 24 vs line 6, a difference of  $74.39 - 70.95 = 3.44$ ). The gap between best 250 result and best ALL result is smaller for the other five TDs (although only slightly smaller for email) and for each of the five TDs there is an ALL result that is statistically indistinguishable from the best 250 result. For this reason, we choose dist=ALL for the extended experiment. Simply using ALL indicator words also has the advantage of eliminating the need to optimize an additional parameter, the number of indicator words selected.

In a way similar to distributional features, the behaviors of WEB and BIO TDs also diverge for dimensionality reduction. The top three results for the WEB TDs are always achieved without SVD (lines 1, 3, 13, 14, 15, 19, 20), the top three results for the BIO TD are all SVD results (lines 6, 22, 24). We opt for the simpler option (no SVD) for the extended experiment in the absence of strong consistent cross-TD evidence for the need of dimensionality reduction. We will also see in the extended experiment that we can recover and surpass the best BIO result (74.39, line 24) by optimizing other parameters.

The results on weighting argue against using binary weighting: the six best results in the table all use tf weighting, either by itself or in conjunction with idf (lines 3, 14, 24). Apparently, the distinction between lower and higher frequencies of indicator word occurrences is beneficial for unknown word classification. Whether tf or tf-idf is better, is less clear. For two TDs, tf-idf yields the best result (grp on line 3, BIO on line 24), for four TDs tf (rev, blog, answer, email: line 14). The difference between best tf-idf and best tf result is not significant for grp; we will get tf results for BIO that are better than the best tf-idf result of 74.39 in Table 1. For this reason, we choose the setting tf for the extended experiment. Again, we are selecting the simpler of two options (tf vs tf-idf) when faced with somewhat mixed evidence.

In summary, based on the results of the base experiment, we choose the following settings for the extended experiment: shape = ALL, dist = ALL, svd = n, wght = tf. For shape, dist, and svd this is the simpler of two possible settings. For

weighting, we choose tf (instead of the simpler binary option) because of clear evidence that some form of frequency weighting is beneficial across TDs. These settings correspond to line 20 in Table 1. This line is repeated as the baseline on line 1 in Table 2. Admittedly, choosing this as a baseline setting is somewhat arbitrary as one could always weigh the optimization criteria – peak performance, robustness, simplicity – differently.

	grp	rev	blog	ans'r	em'l	BIO
1 baseline	56.62	<u>64.61</u>	<u>71.86</u>	54.28	61.05 <sup>†</sup>	65.64 <sup>†</sup>
2 CRF	<u>58.18</u>	64.51	70.48	<u>56.52</u>	<u>63.10</u>	56.62 <sup>†</sup>
3 SD+TD-R	55.50	64.13	<u>72.50</u>	<u>55.31</u>	62.91	65.17 <sup>†</sup>
4 no case NRM	52.83 <sup>†</sup>	64.45	70.68	52.00 <sup>†</sup>	59.27 <sup>†</sup>	67.51 <sup>†</sup>
5 digit NRM	<u>56.80</u>	<u>64.61</u>	<u>72.01</u>	54.05	<u>63.88</u>	68.61 <sup>†</sup>
6 shape only, ALL	48.77 <sup>†</sup>	45.32 <sup>†</sup>	56.58 <sup>†</sup>	39.90 <sup>†</sup>	49.19 <sup>†</sup>	52.52 <sup>†</sup>
7 shape only, 100	47.69 <sup>†</sup>	39.16 <sup>†</sup>	51.90 <sup>†</sup>	36.17 <sup>†</sup>	47.24 <sup>†</sup>	50.14 <sup>†</sup>
8 dist only, ALL	52.05 <sup>†</sup>	63.34	68.21 <sup>†</sup>	47.07 <sup>†</sup>	53.06 <sup>†</sup>	73.41 <sup>†</sup>
9 dist only, 250	51.49 <sup>†</sup>	64.13	66.34 <sup>†</sup>	45.76 <sup>†</sup>	54.13 <sup>†</sup>	72.86 <sup>†</sup>
10  w  > 1	56.58	<u>64.67</u>	71.81	<u>54.84</u>	60.83 <sup>†</sup>	65.99 <sup>†</sup>
11  w  > 2	<u>57.06</u>	<u>64.61</u>	71.56	54.38	<u>63.17</u>	68.61 <sup>†</sup>
12  w  > 3	55.33	60.89 <sup>†</sup>	69.69	48.79 <sup>†</sup>	62.39	73.84 <sup>†</sup>
13  w  > 4	52.87 <sup>†</sup>	60.10 <sup>†</sup>	67.67 <sup>†</sup>	47.53 <sup>†</sup>	53.06 <sup>†</sup>	77.66
14  w  > 5	53.09 <sup>†</sup>	59.35 <sup>†</sup>	66.58 <sup>†</sup>	44.37 <sup>†</sup>	51.69 <sup>†</sup>	77.66
15  w  > 6	52.27 <sup>†</sup>	58.55 <sup>†</sup>	66.93 <sup>†</sup>	43.25 <sup>†</sup>	49.74 <sup>†</sup>	<u>77.74</u>
16  w  > 7	51.96 <sup>†</sup>	56.64 <sup>†</sup>	63.18 <sup>†</sup>	40.46 <sup>†</sup>	47.17 <sup>†</sup>	<u>78.41</u>
17  w  > 8	49.59 <sup>†</sup>	56.16 <sup>†</sup>	58.26 <sup>†</sup>	39.06 <sup>†</sup>	44.31 <sup>†</sup>	<u>79.77</u>
18  w  > 9	46.87 <sup>†</sup>	52.82 <sup>†</sup>	55.54 <sup>†</sup>	33.94 <sup>†</sup>	42.69 <sup>†</sup>	<u>74.58</u> <sup>†</sup>
19  w  > 10	43.42 <sup>†</sup>	51.22 <sup>†</sup>	52.54 <sup>†</sup>	33.33 <sup>†</sup>	39.24 <sup>†</sup>	76.10 <sup>†</sup>

Table 2: Extended experiment. The effect of various parameter changes on accuracy of unknown word classification. “NRM” = “normalization.

**Extended experiment.** In the extended experiment, we investigate the effect of additional parameters. Results are shown in Table 2. Underlining conventions and statistical test setup are identical to Table 1. The CRF baseline used a parameter setting similar to word classification with two exceptions: we set dist=250 because we were not able to run dist=ALL due to memory limitations; and we convert all features to binary due to space restrictions.

Using sequence classification instead of word classification for unknown word prediction does not consistently improve results (line 2). For grp and answer, the CRF achieves the best overall accuracy, but the difference to the baseline is not significant. For the other four TDs, the best result occurs in a different parameter setting. For BIO, a large drop in performance occurs (from 65.64 to 56.62), perhaps suggesting that word classification is more robust than sequence classification for unknown words.

Calculating distributional vectors on both source and target (as opposed to target only) has similarly inconsistent effects (line 3). Perfor-

mance compared to the baseline decreases for four TDs and increases for two. Based on this evidence, SD distributional information is not robust cross-TD and should probably not be used.

Omitting case normalization (line 4) consistently hurts for WEB TDs, but helps for BIO. In other words, for BIO it is better not to case-normalize words. This result is plausible because case conventions vary considerably in different TDs. Whether keeping case distinctions is helpful or not depends on how similar source and target are in this respect and is therefore not stable in its effect across TDs.

Digit normalization (line 5) has a minor positive or negative effect on the first four TDs, but increases accuracy by more than 2% in the last two, email and BIO. The makeup of the WSJ tag set makes it unlikely that differences between digits could result in POS differences that are predictable in unsupervised DA. This argues for using digit normalization when WSJ is the SD.

The clearest result of the table is that distributional information is necessary for good performance. Performance compared to the baseline drops in all cases and all accuracies on lines 6&7 are significantly worse than the best result. Moreover, distributional features seem to encode more meaningful information for POS tagging than shape features; results on lines 6&7 are consistently lower than results on lines 8&9.

The evaluation is similarly consistent for shape information in the WEB TDs (lines 8 and 9). All accuracies are below the baseline, with some of the drops being quite large, e.g., about 7% for answer and email. Surprisingly, omitting shape information results in a large *increase* of accuracy for the BIO TD. We will further investigate this puzzling result below.

Finally, training set filtering – only training the classifier on words above a threshold length  $k$  – is beneficial for all TDs except for blog; and even for blog, moderate filtering has only a negligible negative effect on accuracy (lines 10–11). In principle, the idea of restricting training to longer words because they are most likely to be representative of unknown words seems to be a good one. However, the effect of filtering is sensitive to the threshold length  $k$ . We leave it to future work to find properties of the TD that could be used as diagnostics for finding a good value for  $k$ .

The motivation of splitting the experiments into

basic experiment and extended experiments was to find a stable point in parameter space for the parameters that are most likely to interact and then look at the effect of the remaining parameters using this stable point as starting point. In Table 2, we see that for the WEB TDs, all variations of experimental conditions either hurt performance or produce only small positive changes in accuracy in comparison to the baseline. This is evidence that our strategy of splitting experiments into basic and extended was sound for these TDs.

		BIO
1	baseline	73.41 <sup>†</sup>
3	SD+TD-R	67.94 <sup>†</sup>
4	no case NRM	72.39 <sup>†</sup>
5	digit NRM	74.15 <sup>†</sup>
10	$ w  > 1$	73.96 <sup>†</sup>
11	$ w  > 2$	75.24 <sup>†</sup>
12	$ w  > 3$	81.30 <sup>†</sup>
13	$ w  > 4$	81.88 <sup>†</sup>
14	$ w  > 5$	82.98
15	$ w  > 6$	82.47
16	$ w  > 7$	<u>84.46</u>
17	$ w  > 8$	<u>83.09</u>
18	$ w  > 9$	79.03 <sup>†</sup>
19	$ w  > 10$	80.52 <sup>†</sup>

Table 3: Extended experiment for BIO without shape information. Dist=ALL.

However, the situation for BIO is different. Two parameter changes result in large performance gains for BIO: omitting shape information (increase by 8%, lines 1 vs 8) and filtering out short training words (increase by 14%, lines 1 vs 17). This indicates that the base configuration of the extended experiment is not a good starting point for exploring parameter variation for BIO.

For this reason, we repeat parts of the extended experiment without any shape information. As we would expect, we obtain results for WEB TDs that are consistently worse than those in Table 2 (not shown), with one exception: a slight increase for  $|w| > 8$  in email. However, the results for BIO are much improved as shown in Table 3.

To conclude, we found that shape information is helpful for the WEB TDs, but it decreases performance by about 10% for BIO. We will analyze the reason for this discrepancy in the next section.

As a last set of experiments, we run the optimal parameter combination ( $|w| > 7$  in Table 3, 84.46) on the BIO test set and obtained an accuracy of 88.13. This is more than 10% higher than the best number for unknown word prediction on BIO published up to this point (76.3 by Huang and Yates (2010)). For the experimental conditions with the best WEB results in Table 2

(double underlining), we get the following test accuracies: grp=56.66, rev=67.79, blog=64.80, answer=66.51, email=65.51. These are either better than dev or slightly worse except for blog; the blog result can be explained by the fact that the blog base model (line 1) also is a lot worse on test than on dev (66.08 vs 71.86). We interpret these test set results as indicating that we did not overfit to the development set in our experiments.

**Summary.** We have investigated the cross-TD robustness of a number of configurational choices in DA for POS tagging. Based on our results, the following choices are relatively robust across TDs: using ALL indicator words (as opposed to a subset) for distributional features, no dimensionality reduction, tf weighting, digit normalization, target-only distributional features, and formalization of the problem of unknown word prediction as word classification (as opposed to sequence classification).

We found other choices to be dependent on the TD, in particular the use of shape features, case normalization and training set filtering.

The most important lesson from these results is that many aspects of DA are highly dependent on the TD. Given our results, it is unlikely that a single DA setup will work in general. Instead, criteria need to be developed that allow us to predict which features and methods work for different TDs.

## 4 Analysis and discussion

The biggest TD differences we found in the experiments are those between WEB and BIO: they behave differently with respect to dimensionality reduction (bad for WEB, good for BIO), shape information (good for WEB, bad for BIO) and sequence classification (neutral for WEB, bad for BIO).

One hypothesis that could explain these results is that the difference between BIO and WSJ is larger than the difference between WEB and WSJ. For example, dimensionality reduction creates more generalized representations, which may be appropriate for TDs with large source-target differences like BIO; and WSJ suffixes may not be helpful for BIO because biomedical terminology has suffixes specific to scientific vocabulary and is rare in newspaper text. In contrast, WEB suffixes may not diverge as much from WSJ since both are “non-technical” genres.

One way to assess the difference between two



TD	tags	suffixes	transitions
grp	.009	.275	.068
rev	.057	.352	.212
blog	.009	.295	.074
answer	.048	.337	.158
email	.036	.273	.139
BIO	.096	.496	.385

Table 4: JS divergences between WSJ and TDs.

domains is to compare various characteristic probability distributions. The distance of two domains under a representation  $R$  has been shown to be important for DA (Ben-David et al., 2007). Similar to Huang and Yates (2010), we use Jensen-Shannon (JS) divergence as a measure of divergence. Table 4 shows the JS divergences between WSJ and the six TDs for different distributions.

The results confirm our hypothesis. BIO is indeed more different from WSJ than the other TDs. Tag distribution divergence is 0.096 for BIO and ranges from 0.009 to 0.057 for WEB. Suffix distribution divergence of BIO is 0.496, almost 50% more than rev, the WEB TD with highest suffix divergence. The underlying probability distributions here are  $P(\text{suffix}|t)$ , where  $t \in \{\text{NN}, \text{NNP}, \text{JJ}\}$  – most unknown words are in these three classes and accuracy is therefore mostly a measure of accuracy on NN, NNP and JJ. Finally, transition probability divergence of BIO for NN, NNP, JJ is also much larger than for WEB. The distribution investigated here is  $P(t_{i-1}|t_i)$ ; we compute the divergence between, say, BIO and WSJ for the three tags and then average the three divergences.

We do not have space to show detailed results on all tags, but the divergences are more similar for closed class POS. E.g., there is virtually no difference in transition probability divergence for modals between BIO and WEB. This observation prompted us to investigate whether some TD differences might depend on the evaluation measure used. Accuracy – a type of microaveraging – is mostly an evaluation of the classes that are frequent for unknown words: NN, NNP, JJ. If most of the higher divergence of BIO is caused by these categories, then a macroaveraged evaluation, which gives equal weight to each POS tag, should show less divergence.

This is indeed the case as the macroaveraged results in Table 4 show. These results are more consistent across TDs than those evaluated with accuracy. Removing shape and distributional information now hurts performance for all TDs (lines

		grp	rev	blog	ans'r	em'l	BIO
1	baseline	32.77	38.89	43.48	30.52	34.26	40.06
2	CRF	38.74	42.71	46.63	38.08	36.21	39.03
3	SD+TD-R	<u>32.87</u>	<u>38.55</u>	<u>44.75</u>	<u>33.19</u>	<u>35.30</u>	<u>41.42</u>
4	no case NRM	27.08	39.82	39.54	25.80	27.33	39.98
5	digit NRM	32.80	39.09	43.68	30.47	34.69	37.72
6	shape only, ALL	18.02	21.25	24.61	16.25	16.37	26.55
8	dist only, ALL	27.70	38.39	34.38	22.11	29.71	37.01
10	$ w  > 1$	32.73	<u>39.48</u>	43.54	<u>30.60</u>	34.20	35.32
11	$ w  > 2$	<u>33.33</u>	37.38	43.52	30.02	34.66	35.05
13	$ w  > 4$	26.37	28.92	37.68	22.33	24.14	37.55

Table 5: Selected conditions of the extended experiment (Table 2), evaluated using macroaveraged  $F_1$ .

6&8). WEB and BIO behave more similarly with respect to training set filtering: the large outliers for BIO we obtained in the accuracy evaluation are gone. SD distributional information has a more beneficial effect on  $F_1$  than on accuracy, probably because the classification of POS that are more stable across TDs like verbs and adverbs benefits from SD information. The CRF produces the best result for all WEB TDs. For less frequent POS classes (those that dominate the macroaveraged measure, especially verbal POS), sequence information and “long-distance” context is probably more stable and can be exploited better than for NN, NNP and JJ. However, there is still a drop-off from the baseline for BIO; we attribute this to the larger differences in the transition probabilities for BIO vs WEB (Table 4); the sequence classifier is at a disadvantage for BIO, even on a macroaveraged measure, because the transition probabilities change a lot.

It is important to note that even though  $F_1$  results are more consistent for DA, accuracy is the appropriate measure to use for POS tagging: the usefulness of a tagger to downstream components in the processing pipeline is better assessed by accuracy than by  $F_1$ .

## 5 Related work

Most work on POS tagging takes a standard supervised approach and assumes that source and target are the same (e.g., (Toutanova et al., 2003)). At the other end of the spectrum is the unsupervised setting (e.g., (Schütze, 1995; Goldwater and Griffiths, 2007)). Other researchers have addressed the task of adapting a known tagging dictionary to a TD (e.g., (Merialdo, 1994; Smith and Eisner, 2005)), which we view as complementary to methods for words about whose tags nothing is known. Subramanya et al. (2010) perform DA without using any unlabeled TD text. All of these applica-

tions scenarios are reasonable; however, it can be argued that the scenario we address is – apart from standard supervised learning – perhaps more typical of what occurs in practice: there is labeled SD text available for training; there is plenty of unlabeled TD text available; and there is a substantial number of TD words that do not occur in the SD. Frequently, researchers make the assumption that a small labeled target text has been created (e.g., (Daumé III, 2007)); in the process, a small number of unknown words may also be labeled, but this is not an alternative to handling unknown words in general.

Work by Das and Petrov (2011) is also a form of DA for POS tagging, using universal POS tag sets and parallel corpora. It is likely that best performance for TDs without training data can be achieved by combining our approach with a multilingual approach if appropriate parallel data is available. Ganchev et al. (2012) use another source of additional information, search logs. Again, it should be possible to integrate search-log based features into our framework.

Blitzer et al. (2006) learn correspondences between features in source and target. Our results suggest that completely ignoring source features (and only using source labels) may be a more robust approach for unknown words.

Cholakov et al. (2011) point out that improving tagging accuracy does not necessarily improve the performance of downstream elements of the processing pipeline. However, improved unknown word classification will have a positive impact on most downstream components.

Choi and Palmer (2012) perform DA by training two separate models on the available data, a generalized one and a domain-specific one. During tagging, an input sentence is tagged by the model that is most similar to the sentence. Since their approach is not conditioned on the underlying tagging model, it would be interesting to integrate their approach with ours.

Huang and Yates (2009) evaluate CRFs with distributional features. Besides raw feature vectors, they examine lower dimensional feature representations using SVD or a special HMM-based method. In our experiments, we did not find an advantage to using SVD.

Huang and Yates (2010) use sequence labeling to predict POS of unknown words. Huang and Yates (2012) extend this work by inducing latent

states that are shown to improve prediction. As we argued above, a word classification approach has several advantages compared to a sequence labeling approach. Since latent sequence states can be viewed as a form of dimensionality reduction, it would be interesting to compare them to the non-sequence-based dimensionality reduction (SVD) we have investigated in our experiments.

Zhang and Kordoni (2006) use a classification approach for predicting POS for in-domain unknown words. They achieve an accuracy of 61.3%. Due to differences in the data sets used, these results are not directly comparable with ours.

Miller et al. (2007) and Cucerzan and Yarowsky (2000) have both investigated the use of suffixes for DA. Miller et al. characterized words by a list of hand-built suffix classes that they appear in. They then used a 5-NN classifier along with a custom similarity measure to find initial lexical probabilities for all words. We also ran extensive experiments with kNN, but found that one-vs-one SVM performs better.

Cucerzan and Yarowsky (2000) use distribution as a backoff strategy if no helpful suffix information is available. They address unknown word prediction for new languages. We have found that for within-language prediction, distributional information is generally more robust than shape information, including suffixes.

Van Asch and Daelemans (2010) find that DA performance is the higher, the more similar the unigram distribution of the TD is to that of the SD. However, we cannot compute unigram distributions in the case of unknown words.

## 6 Conclusions and Future Work

In this paper, we have investigated the robustness of DA representations and methods for POS tagging and shown that there are large differences in robustness across TDs that need to be taken into account when performing DA for a TD. We found that the divergence between source and target is an important predictor of what elements of DA will work; e.g., higher divergence makes it more likely that generalization mechanisms like dimensionality reduction will be beneficial.

In future work, we would like to develop statistical measures of source-target divergence that accurately predict whether a feature type or DA technique supports high-performance DA for a particular TD.

## References

- Shai Ben-David, John Blitzer, Koby Crammer, and Marina Sokolova. 2007. Analysis of representations for domain adaptation. In *NIPS 19*, pages 137–144.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of the EMNLP*, pages 120–128.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Jinho D. Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proc. of the ACL: Short Papers - Vol. 2*, pages 363–367.
- Kostadin Cholakov, Gertjan van Noord, Valia Kordoni, and Yi Zhang. 2011. An empirical comparison of unknown word prediction methods. In *Proc. of the IJCNLP*, pages 767–775.
- Silviu Cucerzan and David Yarowsky. 2000. Language independent, minimally supervised induction of lexical probabilities. In *Proc. of the ACL*, pages 270–277.
- Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proc. of the ACL*, pages 600–609.
- Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proc. of the ACL*, pages 256–263.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41(6):391–407.
- Kuzman Ganchev, Keith Hall, Ryan McDonald, and Slav Petrov. 2012. Using search-logs to improve query tagging. In *Proc. of the ACL: Short Papers - Vol. 2*, pages 238–242.
- Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proc. of the ACL*, pages 744–751.
- Fei Huang and Alexander Yates. 2009. Distributional representations for handling sparsity in supervised sequence-labeling. In *Proc. of the Joint Conf. of the ACL and the IJCNLP*, pages 495–503.
- Fei Huang and Alexander Yates. 2010. Exploring representation-learning approaches to domain adaptation. In *Proc. of the DANLP Workshop*, pages 23–30.
- Fei Huang and Alexander Yates. 2012. Biased representation learning for domain adaptation. In *Proc. of the EMNLP-CoNLL*, pages 1313–1323.
- Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The Penn treebank. *Comp. Linguistics*, 19(2):313–330.
- Bernard Merialdo. 1994. Tagging english text with a probabilistic model. *Comp. Linguistics*, 20(2):155–171.
- John Miller, Manabu Torii, and Vijay K. Shanker. 2007. Building domain-specific taggers without annotated (domain) data. In *Proc. of the EMNLP-CoNLL*, pages 1103–1111.
- Thomas Müller, Hinrich Schütze, and Helmut Schmid. 2012. A comparative investigation of morphological language modeling for the languages of the european union. In *Proc. of the NAACL-HLT*, pages 386–395.
- Naoaki Okazaki. 2007. CRFsuite: A fast implementation of conditional random fields (CRFs). Available at: <http://www.chokkan.org/software/crfsuite/>.
- Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web. Notes of the 1st SANCL Workshop.
- Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proc. of the EACL*, pages 141–148.
- Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: training log-linear models on unlabeled data. In *Proc. of the ACL*, pages 354–362.
- Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proc. of the EMNLP*, pages 167–176.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of the NAACL-HLT - Vol. 1*, pages 173–180.
- Vincent Van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proc. of the DANLP Workshop*, pages 31–36.
- Yi Zhang and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open texts processing. In *Proc. of the LREC*, pages 275–280.

# Dependency Parsing for Identifying Hungarian Light Verb Constructions

Veronika Vincze<sup>1,2</sup>, János Zsibrita<sup>2</sup> and István Nagy T.<sup>2</sup>

<sup>1</sup>Hungarian Academy of Sciences, Research Group on Artificial Intelligence

vinczev@inf.u-szeged.hu

<sup>2</sup>Department of Informatics, University of Szeged

{zsibrita,nistvan}@inf.u-szeged.hu

## Abstract

Light verb constructions (LVCs) are verb and noun combinations in which the verb has lost its meaning to some degree and the noun is used in one of its original senses. They often share their syntactic pattern with other constructions (e.g. verb-object pairs) thus LVC detection can be viewed as classifying certain syntactic patterns as light verb constructions or not. In this paper, we explore a novel way to detect LVCs in texts: we apply a dependency parser to carry out the task. We present our experiments on a Hungarian treebank, which has been manually annotated for dependency relations and light verb constructions. Our results outperformed those achieved by state-of-the-art techniques for Hungarian LVC detection, especially due to the high precision and the treatment of long-distance dependencies.

## 1 Introduction

Multiword expressions (MWEs) are lexical items that can be decomposed into single words and display lexical, syntactic, semantic, pragmatic and/or statistical idiosyncrasy (Kim, 2008). Light verb constructions (LVCs) form a subtype of MWEs: they are verb and noun combinations in which the verb has lost its meaning to some degree and the noun is used in one of its original senses (e.g. *make a decision* or *take a walk*). In several NLP applications like information retrieval or machine translation it is important to identify LVCs in context since they require special treatment, particularly because of their semantic features. Thus, LVCs should be identified to help these applications.

Light verb constructions (e.g. *make a mistake*) often share their syntactic pattern with literal verb + noun combinations (e.g. *make a cake*). Thus,

specific syntactic constructions – e.g. verb-object pairs – can be separated into two classes: one where the noun behaves as a real object (*cake*) and one where the noun functions as the light verb object (*mistake*). Thus, LVC detection can be viewed as classifying certain syntactic patterns as LVCs or not and assigning a specific syntactic label to the argument of the light verb.

In this paper, we explore a novel way to LVC detection: we apply a dependency parser to carry out the task. Although the usability of identified multiword expressions has been investigated in the literature (see Section 4), and many MWE detection systems rely on syntactic information, we are not aware of any approach that aimed at applying a dependency parser for the dedicated task of identifying LVCs. Our approach requires a treebank annotated for syntactic and LVC information at the same time. Due to the availability of annotated resources, we focus on light verb constructions in Hungarian, a morphologically rich language. Thus, we present our experiments on the legal subcorpus of the Szeged Dependency Treebank annotated for LVCs (Vincze and Csirik, 2010) as well as dependency relations (Vincze et al., 2010). We will pay special attention to non-contiguous LVCs in our investigations as there are quite a few non-contiguous LVCs in Hungarian due to the free word order. Our results empirically prove that LVCs can be detected as a “side effect” of dependency parsing.

## 2 Light Verb Constructions in Hungarian

Hungarian is an agglutinative language, which means that a word can have hundreds of word forms due to inflectional or derivational morphology (É. Kiss, 2002). Hungarian word order is related to information structure, e.g. new (or emphatic) information (focus) always precedes the verb and old information (topic) precedes the

focus position. Thus, the position relative to the verb has no predictive force as regards the syntactic function of the given argument. In English, the noun phrase before the verb is most typically the subject whereas in Hungarian, it is the focus of the sentence, which itself can be the subject, object or any other argument.

The grammatical function of words is determined by case suffixes. Hungarian nouns can have about 20 cases, which mark the relationship between the verb and its arguments (subject, object, dative etc.) and adjuncts (mostly adverbial modifiers). Although there are postpositions in Hungarian, case suffixes can also express relations that are expressed by prepositions in English. Verbs are inflected for person and number and the definiteness of the object. There are several other linguistic phenomena that are syntactic in nature in English but they are encoded morphologically in Hungarian. For instance, causation and modality are expressed by derivational suffixes.

The canonical form of a Hungarian light verb construction is a bare noun + third person singular verb, for instance, *tanácsot ad* advice-ACC give “to give advice”. Due to the above features, they may occur in non-canonical versions as well: the verb may precede the noun, or they may be not adjacent, moreover, the verb may occur in different surface forms inflected for tense, mood, person and number.

LVCs may occur in several forms due to their syntactic flexibility. Besides the prototypical verbal form in Hungarian, they can have a participial form (e.g. *figyelembe vevő* account-INE taking “taking into account”) and they may also undergo nominalization, yielding a nominal compound (e.g. *életbe lépés* life-INE step “entering into force”).<sup>1</sup>

From a morphological perspective, LVCs can also be divided into groups. First, the nominal component is the object of the verb, i.e. it bears an accusative case in Hungarian (e.g. *döntést hoz* decision-ACC bring “to make a decision” or *tanácsot ad* advice-ACC give “to give advice”). Second, the nominal component can bear other (oblique) cases as well (e.g. *zavarba*

<sup>1</sup>Due to some orthographical rules, certain nominal or participial occurrences of LVCs should be spelt as one word in Hungarian (such as *tanácsadó* advice.giver “consultant”). These latter cases are not identifiable with syntax-based methods, only with morphological methods, thus we omit them from our investigations.

*hoz* embarrassment-ILL bring “to embarrass” or *figyelemmel kísér* attention-INS follow “to pay attention”). Third, – although rarely – a postpositional phrase can also occur in the construction (e.g. *uralom alá jut* rule under get “to get under rule” or *hatás alatt áll* effect under stand “to be under effect”).

### 3 Light Verb Constructions as Complex Predicates

Although light verb constructions are made of two parts, namely, the nominal component and the verb, thus, they show phrasal properties, it can be argued that from a semantic point of view they form one unit. First, many light verb constructions have a verbal counterpart with the same meaning (e.g. *döntést hoz* decision-ACC bring “to make a decision” – *dönt* “to decide”). Second, there are meanings that can only be expressed through a light verb construction (e.g. *házkutatást tart* (search.of.premises-ACC hold) ‘to conduct search of premises’ in Hungarian). Third, there are languages that abound in verb + noun constructions or multiword verbs (such as Estonian (Muischnek and Kaalep, 2010) or Persian (Mansoori and Bijankhan, 2008)): verbal concepts are mostly expressed by combining a noun with a light verb (Mansoori and Bijankhan, 2008).

On the other hand, there are views that the relationship between the verbal and the nominal component is not that of a normal argument. For instance, Meyers et al. (2004) assume that support verbs (a term related to light verbs) share their arguments with a noun. Chomsky (1981, p.37) calls *advantage* a quasi-argument of *take* in the idiom *take advantage of*.<sup>2</sup> Alonso Ramos (1998) proposes the role of quasi-object: this relationship holds between parts of idiomatic constructions, which is in accordance with Chomsky’s usage of the term *idiom*. In this spirit, the term *quasi-argument* might be extended to signal the relationship between the verbal and the nominal components of light verb constructions as well since they behave as a semantic unit, forming one complex predicate.

Higher-level NLP applications can also profit from this solution because the identification of light verb constructions can be enhanced in this way, which has impact on e.g. information extrac-

<sup>2</sup>In our view, *take advantage of* is a light verb construction rather than an idiom.

tion (IE). For instance, in event extraction the parser should recognize the special status of the quasi-argument and treat it in a specific way as in the following sentence:

Pete **made a decision** on his future.

Thus, the following data can be yielded by the IE algorithm:

EVENT: decision-making  
ARGUMENT<sub>1</sub>: Pete  
ARGUMENT<sub>2</sub>: his future

Instead of:

\*EVENT: making  
ARGUMENT<sub>1</sub>: Pete  
ARGUMENT<sub>2</sub>: decision  
ARGUMENT<sub>3</sub>: his future

Thus, there is an event of **decision-making**, **Pete** is its subject and it is about **his future** (and not an event of **making** with the arguments **decision**, **Pete** and **his future** as it would be assumed if *decision* was not marked as a quasi-argument of the verb).

In order to reach this way of representation, there are two possibilities. First, we employ linguistic preprocessing of the data (including dependency parsing), then an LVC detector is used and in a post-processing step after syntactic parsing, the special relation of the nominal and the verbal component should be marked, i.e. certain syntactic labels are overwritten. Second, we execute parsing in a way that the training dataset already contains LVC-specific syntactic labels, that is, it is the dependency parser that carries out LVC detection. In this paper, we experiment with both ways and present and evaluate our results.

## 4 Related Work

There have been a considerable number of studies on LVC detection for several languages. They have been automatically identified in several languages such as English (Cook et al., 2007; Tu and Roth, 2011), Dutch (Van de Cruys and Moirón, 2007), Basque (Gurrutxaga and Alegria, 2011) and German (Evert and Kermes, 2003) just to mention a few.

We are aware of one machine learning system that identifies Hungarian LVCs in texts: the system described in Vincze et al. (2013) selects LVC

candidates from texts on the basis of syntactic information, then in a second step it classifies them as genuine LVCs or not, using morphological, lexical, syntactic and semantic features.

Regarding the methods they use, Fazly and Stevenson (2007), Van de Cruys and Moirón (2007) and Gurrutxaga and Alegria (2011) used statistical features for identifying LVCs. Others employed rule-based systems (Diab and Bhutada, 2009; Nagy T. et al., 2011), which usually make use of (shallow) linguistic information. Some hybrid systems integrated both statistical and linguistic information as well (Tan et al., 2006; Tu and Roth, 2011).

As we aim at identifying LVCs by applying a dependency parser, next we concentrate on studies that are based on syntactic information and are related to MWE extraction. Seretan (2011) developed a method for collocation extraction based on syntactic constraints. Wehrli et al. (2010) argued that collocations can highly contribute to the performance of the parser since many parsing ambiguities can be excluded if collocations are known and treated as one syntactic unit. Nivre and Nilsson (2004) analyzed the influence of (previous) MWE recognition on dependency parsing and showed that known MWEs have a beneficial effect on parsing results. Korkontzelos and Manandhar (2010) investigated whether known MWEs improve the performance of statistical shallow parsers and found that they can significantly contribute to the efficiency of parsing. Eryiğit et al. (2011) analysed the impact of extracting MWEs on improving the accuracy of a dependency parser in Turkish. They found that the integration of compound verb and noun formations (which concept is similar to the one of light verb constructions applied here) has a detrimental effect on parsing accuracy since it increases lexical sparsity.

As can be seen, many previous studies examined the effects of already identified MWEs on the efficiency of parsing. On the other hand, there have been some current studies that aim at experimenting in the other direction, namely, using parsers for identifying MWEs: constituency parsing models are employed in identifying contiguous MWEs in French and Arabic (Green et al., 2013). Their method relied on a syntactic treebank, an MWE list and a morphological analyzer.

In this paper, we also experiment in this area: we employ a dependency parser for identifying

LVCs in Hungarian texts as a “side effect” of parsing sentences. Our dependency parser based method for identifying Hungarian LVCs is novel since to the best of our knowledge, dependency parsers have not been directly applied to identify LVCs. Moreover, it requires only a syntactic treebank enhanced with LVC annotation, in other words, there is no need to implement a separate LVC detector from scratch. In the following, we present our experiments and discuss our results.

## 5 Experiments

In this section, we will present our corpus, our methodology for detecting light verb constructions and we will show our results.

### 5.1 The Corpus

The Szeged Constituency Treebank has been manually annotated for light verb constructions (Vincze and Csirik, 2010). This treebank exists in another manually annotated version, namely, with dependency annotation (Vincze et al., 2010). Thus, manual annotations for LVCs and dependency structures are available for the same bunch of texts, which made it possible to map the two manual annotations. Thus, dependency relations were enhanced with LVC-specific relations that can be found between the two members of the constructions. For instance, instead of the traditional OBJ (object) relation, which occurred in the original version of the Szeged Dependency Treebank, the relation OBJ-LVC can be found between the words *döntést* (decision-ACC) and *hoz* “bring”, members of the LVC *döntést hoz* “to make a decision” in the version used in this experiment. Here we provide a list of LVC-specific relations that occurred in our data (neglecting a handful of cases which were mislabeled due to some annotation errors in the dependency treebank):

- ATT-LVC – relation between a noun and a participial occurrence of a light verb:

*(a tegnapi) adott tanács*

(the yesterday) given advice

“(the) advice that was given (yesterday)”

- OBJ-LVC – relation between a light verb and its object:

*bejelentést tesz*

announcement-ACC makes

“to make an announcement”

- OBL-LVC – relation between a light verb and its nominal argument (which is not the subject or object or dative):

*életbe lép*

life-ILL step

“to take effect”

- SUBJ-LVC – relation between a light verb and its subject:

*sor kerül (vmire)*

turn get sg-SUB

“the time has come for sg”

When mapping the LVC annotations and the dependency structures, we paid attention to the fact that it is only LVCs spelt as two tokens that could be identified with our methodology since no internal structure of compound words are marked in the Hungarian treebank and thus no dependency relation can be found among the members of the compound. So, we neglect LVCs spelt as one word and focus only on verbal and participial LVCs that consist of two members (cf. Footnote 1).

Figure 1 shows an example of a sentence with and without LVC-specific dependency labels. As can be seen, we have the light verb construction *döntést hoz* decision-ACC bring “to make a decision” in the sentence. However, it is parsed as a “normal” object of the verb in the first case (OBJ) and as a light verb object (OBJ-LVC) in the second case. Moreover, it is also seen that the two components of the LVC are not adjacent hence there are crossing branches in the dependency graph.

Although the entire Szeged Corpus contains manual LVC and dependency annotation, for the purpose of our study, we just selected texts from the law domain since they contain the biggest number of LVCs. Sentences in the law subcorpus were further filtered due to the fact that state-of-the-art dependency parsers cannot adequately treat verbless sentences, hence verbless sentences were ignored (see Farkas et al. (2012) for a detailed discussion of the problem). After this filtering step, we experimented with 6173 sentences, which consist of 156,744 tokens and contain 1101 LVCs. We present statistical data on the frequency of the LVC-specific relations in Table 1.

As Hungarian is a free word order language, the two components of LVCs, namely, the noun and the light verb, may not be adjacent in all cases,

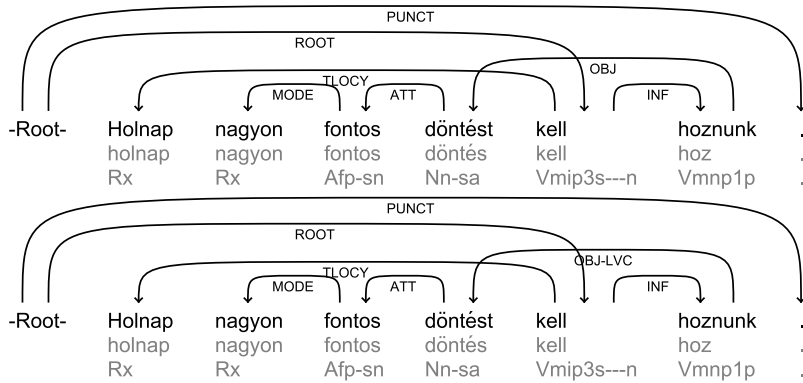


Figure 1: Dependency graph of the sentence *Holnap nagyon fontos döntést kell hoznunk* “Tomorrow we will have to make a very important decision” with or without LVC-specific dependency relations.

Relation	#	Non-contiguous	%
ATT-LVC	142	60	42.3
OBJ-LVC	587	231	39.4
OBL-LVC	266	50	18.8
SUBJ-LVC	102	4	3.9
Other-LVC	4	2	50
Total	1101	347	31.5

Table 1: Distribution of relations in the gold standard data and the frequency of non-contiguous LVCs.

which has a potentially detrimental effect on their identification in texts. Thus, we investigated the frequency of such cases in the data. Table 1 reveals that it is a quite frequent phenomenon in the corpus: almost one third of LVCs are non-contiguous. The largest distance between the noun and the verb is 21 tokens and the average distance between the two non-adjacent components is 4.28 tokens. All this suggests that sequence labeling approaches for LVC detection may not be as effective on the data as expected, however, a dependency parser that is able to identify long-distance dependencies may deal with the problem of non-adjacent but grammatically dependent elements in a more accurate way, which we will test below.

## 5.2 Dependency Parsing for LVC Detection

Farkas et al. (2012) carried out the first experiments on Hungarian dependency parsing. They empirically showed that state-of-the-art dependency parsers achieve similar results – in terms of attachment scores – on Hungarian and English.

Although the results are not directly comparable due to domain differences and annotation schema divergences, they concluded that the difficulty of parsing Hungarian is very similar to parsing English and statistical dependency parsing is a viable way of parsing Hungarian, a morphologically rich language with free word order.

As their results indicated, the Bohnet dependency parser (Bohnet, 2010) proved to be the most effective on Hungarian data (Farkas et al., 2012), thus we applied it in our experiments too. It is an efficient second order dependency parser that models the interaction between siblings as well as grandchildren. Its decoder works on labeled edges, i.e. it uses a single-step approach for obtaining labeled dependency trees. It uses a rich and well-engineered feature set and it is enhanced by a Hash Kernel, which leads to higher accuracy.

Due to the free word order, there are quite many long-distance dependencies in Hungarian sentences, where a word and its parent are not adjacent (see also Figure 1). However, these linguistic phenomena are reasonably well-treated by dependency parsers. Furthermore, there seem to be quite a lot of non-contiguous LVCs in Hungarian. Hence, we think that these facts justify our experiments on applying a dependency parser for identifying LVCs.

## 5.3 Methodology

We trained and evaluated the Bohnet parser on the data in a ten-fold cross validation manner. To evaluate the quality of the dependency parsing, we applied the Labeled Attachment Score (LAS) and Unlabeled Attachment Score (ULA) metrics, taking into account punctuation as well. On the other hand, we also employed  $F_{\beta=1}$  scores inter-



Method	Precision	Recall	F-score
Dictionary matching	0.7849	0.1229	0.2125
Classification	0.8284	0.6760	0.7445
Dependency parser	0.8660	0.6712	0.7563

Table 2: Results on LVC detection.

preted on the LVC-specific relations to evaluate the performance of detecting LVCs in the corpus and we evaluated our system on contiguous and non-contiguous LVCs as well.

As baselines, we made use of the methods described in Vincze et al. (2013). They first employed dictionary matching, where LVCs collected from a parallel corpus annotated for Hungarian LVCs (Vincze, 2012) were mapped to the lemmatized texts. We also applied dictionary matching as one of our baselines. The main method of Vincze et al. (2013) first parsed each sentence and extracted potential LVCs on the basis of the dependency relations found between verb-object, verb-subject, verb-prepositional object, verb-other argument and noun-modifier pairs. The dependency labels were provided by `magyarlan` (Zsibrita et al., 2013). Later, C4.5 decision trees were applied to classify candidate LVCs, which exploits a rich feature set. For instance, morphological features exploited the fact that the nominal component of LVCs is typically derived from a verbal stem or coincides with a verb, on the other hand, the POS tags of the words and surrounding words were also used as features. As for semantic features, the `activity` or `event` semantic senses were looked for among the upper level hyperonyms of the head of the noun phrase in the Hungarian WordNet<sup>3</sup>. As lexical features, fifteen typical light verbs were selected from the list of the most frequent verbs taken from the Szeged ParalellFX corpus (Vincze, 2012) and it was checked whether the lemmatized verbal component of the candidate was one of these fifteen verbs. The lemma of the noun was also applied as a lexical feature.

We evaluated our database with this system too in a ten-fold cross validation manner (using the same data splits as previously) and as evaluation metrics, we employed  $F_{\beta=1}$  scores. The results of our experiments are shown in Tables 2 and 3.

Method	Precision	Recall	F-score
Contiguous LVCs			
Classification	0.8746	0.7854	0.8276
Dependency parser	0.9008	0.7357	0.8099
Non-contiguous LVCs			
Classification	0.7103	0.5188	0.6000
Dependency parser	0.7940	0.5362	0.6401

Table 3: Results on detecting contiguous and non-contiguous LVCs.

## 6 Results

As Table 2 shows, the dependency parser with the LVC-specific relations achieved an F-score of 0.7563 (recall: 0.6712, precision: 0.8660) interpreted on the LVC-specific relations. This result exceeds the ones obtained by the baselines: it outperforms the dictionary matching method by 54.38% in terms of F-score with a considerably better recall value, and, on the other hand, it also performs better than the classification method with a 1.18% gain in F-score – the results are significant (ANOVA,  $p = 0.012$ ). In the latter case, the improvement is due to the higher precision value.

The identification of non-contiguous LVCs proved to be more difficult for both methods than that of contiguous LVCs. The classification approach significantly outperforms the dependency parser on the contiguous LVC class (ANOVA,  $p = 0.0455$ ) but on the non-contiguous class the dependency parser performs significantly better with an F-score of 0.6401 (ANOVA,  $p = 0.0343$ ).

In order to analyze the performance in more detail, we compared the precision, recall and F-scores for each LVC-specific label. Data in Table 4 reveal that SUBJ-LVCs are the easiest ones to predict (with both high precision and recall values) and participial uses of LVCs are the most difficult to identify (ATT-LVC) relation between the noun and the participle, mostly due to the low recall value. Although the precision value is rather low in the case of objects (OBJ-LVC), objects and other arguments (OBL-LVC) can be detected reasonably well. Table 5 shows results for (non-)contiguous LVC classes. It is revealed that for OBL-LVCs, there is no substantial difference between contiguous and non-contiguous LVCs but for objects and participial LVCs, the dis-

<sup>3</sup><http://www.inf.u-szeged.hu/rgai/HuWN>

Relation	#	Precision	Recall	F-score
ATT-LVC	142	0.8267	0.4366	0.5714
OBJ-LVC	587	0.8365	0.6712	0.7448
OBL-LVC	266	0.9175	0.7105	0.8008
SUBJ-LVC	102	0.9592	0.9216	0.9400
Other-LVC	4	–	–	–

Table 4: Distribution of relations in the gold standard data and results in terms of precision, recall and F-score as predicted by the dependency parser.

Relation & type	Precision	Recall	F-score
ATT-LVC C	0.9524	0.4878	0.6452
ATT-LVC NC	0.6667	0.3667	0.4731
OBJ-LVC C	0.8535	0.7507	0.7988
OBJ-LVC NC	0.8025	0.5478	0.6512
OBL-LVC C	0.9226	0.7176	0.8073
OBL-LVC NC	0.8947	0.6800	0.7727
SUBJ-LVC C	0.9785	0.9286	0.9529
SUBJ-LVC NC	0.6000	0.7500	0.6667

Table 5: Results in terms of precision, recall and F-score as predicted by the dependency parser for (non-)contiguous (NC/C) LVC classes.

tance between the two components of the LVC has an essential effect on the efficiency.<sup>4</sup>

As for the performance on dependency parsing, we got 90.38 (LAS) and 92.12 (ULA) when training with LVC-specific relations. If these results are compared to those achieved with traditional (i.e. non-LVC-specific) relations, then it is revealed that in the latter case LAS is 90.63, i.e. 0.25 percentage point higher, which can be considered negligible.

## 7 Discussion

As the results show, the dependency parsing approach achieved the best results on LVC detection, especially due to the high precision score. This is probably due to the rich feature set applied by the Bohnet parser. Furthermore, our approach to solve the problem of LVC detection as a classification of syntactic constructions by using a dependency parser is also justified by these results.

A comparison with previous parser-based approach to MWE detection might also prove use-

<sup>4</sup>As there were hardly any non-contiguous SUBJ-LVCs in the dataset, we cannot draw any conclusions on the difficulty level of identifying non-contiguous light verb subjects.

ful. Green et al. (2013) employed constituency parsers to identify contiguous MWEs in French and Arabic. As a main difference between our approach and theirs, we applied a dependency parser for the task of LVC detection, which proved especially effective since we worked with a free word order language, thus we had to deal with non-contiguous LVCs as well. Our dependency parser approach could adequately identify them as well, however, experimenting with a constituency parser will be a possible way to continue our work.

In Hungarian, it sometimes happens that a sequence that looks like an LVC is actually not an LVC in the specific context as in *A dékán újabb előadást tartott szükségesnek* the dean new-COMP presentation-ACC hold-PAST-3SG necessary-DAT “The dean thought that another presentation was necessary”. In other contexts, *előadást tart presentation-ACC hold* “to have a presentation” would most probably function as an LVC. However, in this case we encounter with another fixed grammatical construction of Hungarian, namely, *valamilyennek tart valamit* somewhat-DAT hold something-ACC “to regard something as something”, e.g. *szépnek tartja a lányt* beautiful-DAT hold-3SG-OBJ the girl-ACC “he thinks that the girl is beautiful”. Thus, there is no LVC in the above example, but approaches that heavily build on MWE lexicons may falsely identify this verb-object pair as a light verb object-light verb pair since they hardly consider contextual information. In contrast, dependency parsers have access to information about other dependents of the verb hence they may learn that in such cases the presence of a dative dependent argues against the identification of the verb-object pair as an LVC.

As for the specific LVC-relations, our approach was most successful on LVCs where the noun fulfilled the role of the subject (i.e. it had the relation SUBJ-LVC). This may be attributed to the fact that these LVCs are the least diverse in the corpus: there are only a handful of such types, and each LVC type has several occurrences in the data thus they can be easily identified. On the other hand, participial uses of LVCs (ATT-LVC) were the hardest to detect, which is partly due to their lexical divergence and partly due to the fact that currently adjectives and participles are not distinguished in Hungarian morphological parsing, i.e. they have the same morphological codes. Thus, the parser, which heavily builds on morpho-

logical information, has no chance to learn that it is only participles that tend to occur as parts of LVCs but adjectives do not. A distinction of participles and adjectives in the Hungarian computational morphology would most probably have beneficial effects on identifying LVCs.

Our results empirically prove that a dependency parser may be effectively applied to identify LVCs in free texts, provided that we have a dependency model trained on LVC-specific relations, which itself requires a treebank manually annotated for dependency relations and LVCs. Although the LAS scores are somewhat lower than in the case of LVC-less dependency relations, the task of LVC detection can be also performed by the parser. On the other hand, the classification approach needs a trained dependency model since it classifies LVC candidates selected on the basis of syntactic information. It also uses LVC lists gathered from annotated corpora and in order to denote LVC-specific relations (i.e. quasi-arguments) in the case of complex predicates, an extra post-processing step is needed in the workflow. Thus, the resources needed by the two approaches are the same but with the dependency parsing approach, the implementation of a new LVC-detector from scratch might be saved and complex predicates are provided immediately by the parser. Moreover, another advantage of the dependency parser is that it performs better on non-contiguous LVCs, which are frequent in Hungarian.

We also carried out an error analysis in order to compare the two methods. It was difficult for both the dependency parser and the classifier to recognize rare LVCs or those that included a non-frequent light verb. A typical source of error for the dependency parser was that sometimes an LVC-specific relation was proposed for non-nouns (e.g. adverbs or conjunctions) as well, like in *akár írnia* (either write-INF.3SG) “either he should write”, where *akár* was labeled as an LVC-object of the verb instead of a conjunction. Furthermore, the classifier often made an error in cases where the sentence included an LVC but another argument of the verb was labeled as part of the LVC, e.g. *filmet forgalomba hoz* (film-ACC circulation-INE bring) “to put a film into circulation”, where the gold standard LVC is *forgalomba hoz* “to put something into circulation” but *filmet hoz* “to bring a film” was labeled as a false positive LVC. Since different phenomena proved to be difficult for the

two systems, a possible direction for future work may be to combine the two approaches in order to minimize prediction errors.

Here we experimented with Hungarian, a morphologically rich language. Nevertheless, we believe that the method of applying a dependency parser for LVC detection is not specific to this typological class of languages and it can be employed for any language that has a dependency treebank which contains annotation for LVCs.

## 8 Conclusions

In this paper, we empirically showed that a dependency parser can be employed to detect LVCs in free texts. For this, we used a Hungarian treebank, which has been manually annotated for dependency relations and light verb constructions. Our results outperformed those achieved by state-of-the-art techniques for Hungarian LVC detection and the main advantages of our system is its high precision on the one hand and the adequate treatment of non-contiguous LVCs on the other hand.

The error analysis of the systems applied suggests that since the two systems make errors in different cases, combining them may lead to more precise results. Another possible way of improving the system is to explore methods for the treatment of participial LVCs. Furthermore, as future work we aim at experimenting with the dependency parser in other scenarios (e.g. the newspaper subcorpus of the Szeged Dependency Treebank) in order to make further generalizations on the role of dependency parsing in LVC detection.

## Acknowledgments

This work was supported in part by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013) and by the COST Action PARSEME (IC1207).

## References

- Margarita Alonso Ramos. 1998. *Etude sémantico-syntaxique des constructions à verbe support*. Ph.D. thesis, Université de Montréal, Montreal, Canada.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of Coling 2010*, pages 89–97.
- Noam Chomsky. 1981. *Lectures on Government and Binding*. Foris, Dordrecht.

- Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of MWE 2007*, pages 41–48, Morristown, NJ, USA. ACL.
- Mona Diab and Pravin Bhutada. 2009. Verb Noun Construction MWE Token Classification. In *Proceedings of MWE 2009*, pages 17–22, Singapore, August. ACL.
- Katalin É. Kiss. 2002. *The Syntax of Hungarian*. Cambridge University Press, Cambridge.
- Gülşen Eryiğit, Tugay Ilbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of SPMRL 2011*, pages 45–55, Dublin, Ireland. ACL.
- Stefan Evert and Hannah Kermes. 2003. Experiments on candidate data for collocation extraction. In *Proceedings of EACL 2003*, pages 83–86.
- Richárd Farkas, Veronika Vincze, and Helmut Schmid. 2012. Dependency Parsing of Hungarian: Baseline Results and Challenges. In *Proceedings of EACL 2012*, pages 55–65, Avignon, France, April. ACL.
- Afsaneh Fazly and Suzanne Stevenson. 2007. Distinguishing Subtypes of Multiword Expressions Using Linguistically-Motivated Statistical Measures. In *Proceedings of MWE 2007*, pages 9–16, Prague, Czech Republic, June. ACL.
- Spence Green, Marie-Catherine de Marneffe, and Christopher D. Manning. 2013. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.
- Antton Gurrutxaga and Iñaki Alegria. 2011. Automatic Extraction of NV Expressions in Basque: Basic Issues on Cooccurrence Techniques. In *Proceedings of MWE 2011*, pages 2–7, Portland, Oregon, USA, June. ACL.
- Su Nam Kim. 2008. *Statistical Modeling of Multiword Expressions*. Ph.D. thesis, University of Melbourne, Melbourne.
- Ioannis Korkontzelos and Suresh Manandhar. 2010. Can recognising multiword expressions improve shallow parsing? In *NAACL HLT 2010*, pages 636–644, Los Angeles, California, June. ACL.
- Niloofer Mansoori and Mahmood Bijankhan. 2008. The possible effects of Persian light verb constructions on Persian WordNet. In *Proceedings of GWC 2008*, pages 297–303, Szeged, Hungary, January. University of Szeged.
- Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank Project: An Interim Report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA. ACL.
- Kadri Muischnek and Heiki Jaan Kaalep. 2010. The variability of multi-word verbal expressions in Estonian. *Language Resources and Evaluation*, 44(1-2):115–135.
- István Nagy T., Veronika Vincze, and Gábor Berend. 2011. Domain-dependent identification of multiword expressions. In *Proceedings of RANLP 2011*, Hissar, Bulgaria.
- Joakim Nivre and Jens Nilsson. 2004. Multiword units in syntactic parsing. In *Proceedings of MEMURA 2004*.
- Violeta Seretan. 2011. *Syntax-Based Collocation Extraction*. TSD. Springer, Dordrecht.
- Yee Fan Tan, Min-Yen Kan, and Hang Cui. 2006. Extending corpus-based identification of light verb constructions using a supervised learning framework. In *Proceedings of MWE 2006*, pages 49–56, Trento, Italy, April. ACL.
- Yuancheng Tu and Dan Roth. 2011. Learning English Light Verb Constructions: Contextual or Statistical. In *Proceedings of MWE 2011*, pages 31–39, Portland, Oregon, USA, June. ACL.
- Tim Van de Cruys and Begoña Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of MWE 2007*, pages 25–32, Morristown, NJ, USA. ACL.
- Veronika Vincze and János Csirik. 2010. Hungarian corpus of light verb constructions. In *Proceedings of Coling 2010*, pages 1110–1118, Beijing, China, August. Coling 2010 Organizing Committee.
- Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian Dependency Treebank. In *Proceedings of LREC 2010*, Valletta, Malta, May. ELRA.
- Veronika Vincze, István Nagy T., and Richárd Farkas. 2013. Identifying English and Hungarian Light Verb Constructions: A Contrastive Approach. In *Proceedings of ACL-2013: Short Papers*, Sofia. ACL.
- Veronika Vincze. 2012. Light Verb Constructions in the SzegedParalellFX English–Hungarian Parallel Corpus. In *Proceedings of LREC 2012*, Istanbul, Turkey.
- Eric Wehrli, Violeta Seretan, and Luka Nerima. 2010. Sentence analysis and collocation identification. In *Proceedings of MWE 2010*, pages 28–36, Beijing, China, August. Coling 2010 Organizing Committee.
- János Zsibrita, Veronika Vincze, and Richárd Farkas. 2013. magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In *Proceedings of RANLP-2013*, Hissar, Bulgaria.

# Written Dialog and Social Power: Manifestations of Different Types of Power in Dialog Behavior

**Vinodkumar Prabhakaran**

Department of Computer Science  
Columbia University  
New York, NY  
vinod@cs.columbia.edu

**Owen Rambow**

Center for Computational Learning Systems  
Columbia University  
New York, NY  
rambow@ccls.columbia.edu

## Abstract

Dialog behavior is affected by power relations among the discourse participants. We show that four different types of power relations (hierarchical power, situational power, influence, and power over communication) affect written dialog behavior in different ways. We also present a system that can identify power relations given a written dialog.

## 1 Introduction

The recent increase in online social interactions has triggered great interest in computationally analyzing such interactions to gain insights about the discourse participants (DPs). Within the field of analyzing online interactions, there is a growing interest in finding how social power relations between participants are reflected in the various facets of interactions, and whether the power relations can be detected using computational means (Rowe et al., 2007; Bramsen et al., 2011). More recent work has shown that an analysis of the dialog structure (and not just the message content) helps detecting power relations (Biran et al., 2012; Danescu-Niculescu-Mizil et al., 2012).

Understanding the relation between dialog and power may help in various applications. For example, if a dialog system is engineered to behave appropriately given the user’s expectation of relative power (for different types of power), then the user may experience the interaction with the system as more natural. Turning to dialog analysis rather than generation, we can build a computational system to analyze power relations between participants in an interaction. Such a system could have various applications. Power analysis in online forums and communities could be useful in

determining relevance to a user searching the forum. For example, a user may want to limit his search to posts authored by the DPs with higher power. Power analysis may also aid law enforcement agencies to detect leaders and influencers in suspicious online communities. This is especially useful since the real identities of the members of such communities are often not revealed and their hierarchies may not be available to the law enforcement agencies.

The power differential between the DPs may be based on a multitude of factors such as status, authority, role, knowledge and so on. Early computational approaches to analyzing power in interactions relied solely on static power structures such as corporate hierarchies as the source of the power differential (Rowe et al., 2007; Bramsen et al., 2011). More recent studies have looked into dynamic notions of power as well, such as influence (Biran et al., 2012). However, not much work has been done to understand how different types of power differ in the ways they affect how people interact in dialog.

In this paper, we study four different types of power — hierarchical power, situational power, influence and power over communication. We investigate whether all four social power relations are manifested in dialog behavior; we restrict our attention to written dialog, specifically email exchanged in an American corporation. By “dialog behavior”, we mean the choices a DP makes while engaging in dialog. Dialog behavior includes choices that affect dialog structure, such as the choice of when to participate (e.g., does the DP initiate the dialog?), how much to contribute (e.g., is the DP terse or loquacious?), what sort of contribution to make (e.g., which dialog acts does the DP perform? how does the contribution link

to previous dialog contributions?), and what form the contribution should take (e.g., whether to make an overt display of power). The main contribution of this paper is to show that the four types of power we consider are in fact different from one another and that they affect the DPs' behavior in written dialog in different but predictable ways. We analyze these manifestations in the language as well as the dialog structure of interactions. We also present a system to detect the DPs with one of these types of power from threaded email interactions.

In Section 2, we discuss related work in the field. Section 3-4 presents the data, annotations, and inter-rater agreement studies on the annotations. Section 5 summarizes the dimensions of interactions we analyze. We then present the main contributions of this paper: Section 6 analyzes the variations in the manifestations of power among the four types, and Section 7 describes a system to predict persons with any of the four types of power. We then conclude and discuss future work.

## 2 Related Work

Within the dialog community, researchers have studied notions of control and initiative in dialogs (e.g. (Walker and Whittaker, 1990; Jordan and Di Eugenio, 1997)). Walker and Whittaker (1990) define "control of communication" in terms of whether the discourse participants are providing new, unsolicited information. They use utterance level rules to determine which discourse participant (whether the speaker or the hearer) is in control, and extend it to segments of discourse. Their notion of control differs from our notion of power over communication. They model control locally over discourse segments. What we are interested in (and what our annotations capture) is the possession of controlling power by one (or more) participant(s) across the entire dialog, i.e. how a participant controls the communication in a dialog thread in order to achieve its intended goals. Despite this difference in definition, we show in Section 6 that our notion of power over communication correlates with Walker and Whittaker (1990)'s notion of control over discourse segments. Jordan and Di Eugenio (1997) suggest that "initiative" applies to the level of problem solving, just as "control" applies to the dialog level. We leave the investigation into the relation between initiative and situational power for future work.

In social sciences, different typologies of power have been proposed. Wartenberg (1990) makes the distinction between power-over and power-to in the context of interactions. Power-over refers to relationships between interactants set by external power structures, while power-to refers to the ability an interactant possesses within the interaction, even if it is temporary. Our notions of hierarchical power and influence are special cases of power-over. Hierarchical power is determined by organizational hierarchy, while influence is determined by knowledge, expertise etc. Similarly, our notions of situational power and power over communication are special cases of power-to. Situational power applies to the situation or task at hand, while power over communication applies to the interaction itself. French and Raven (1959) proposed five bases of power: Coercive, Reward, Positional, Referent, and Expert. They are widely used to study power in sociology. We consider hierarchical power, situational power and power over communication to be positional in nature; although the former two can also have bases in coercion and rewards. The bases of influence are mainly referent and expert power.

Studies in sociolinguistics have also explored the relation between dialog behavior and social power. O'Barr (1982) shows that power relations are manifested in language use in courtroom dialogs. Locher (2004) studies politeness in dialogs in relation to the exercise of power. The correlation between discourse structure and perceived influence of participants has also been studied (Ng et al., 1993; Ng et al., 1995). Specifically, factors such as frequency of contribution, proportion of turns, and number of successful interruptions have been identified as important indicators of influence (Reid and Ng, 2000). This work was done entirely on spoken dialog. In our work, we show that the core insight — conversation is a resource for influence — carries over to written dialog; we also show that it carries over to other forms of power. However, some of the characteristics of spoken dialog do not carry over directly to written dialog, most prominently among them the issue of interruptions: there is no interruption in written dialog.

We now look at various computational approaches to extract power relations from online dialogs. Several studies have used Social Network Analysis (e.g., (Rowe et al., 2007)) to extract social relations from online communication.

Researchers have also applied NLP techniques on message content to detect power relations. Earlier approaches used simple lexical features (e.g. (Bramsen et al., 2011; Gilbert, 2012)) while later studies have performed deeper discourse analysis and used features such as linguistic coordination (Danescu-Niculescu-Mizil et al., 2012), language uses such as attempts to persuade and various other dialog patterns (Biran et al., 2012). We present a more detailed discussion of the above mentioned studies and how they differ from our line of research in (Prabhakaran et al., 2012c).

Our research also falls into the category of studies that go beyond pure lexical features and use dialog structure based features to extract social power relations. In (Prabhakaran et al., 2012c), we studied the notion of situational power in depth and presented a system to detect persons with situational power using dialog features. In this paper as well, we use the system described in (Prabhakaran et al., 2012c). However, this work differs from (Prabhakaran et al., 2012c) and other studies described above in that our focus is on how different types of power are manifested differently in the dialog behavior of the participants. We show that the types of power we consider are in fact different and vary in the ways they manifest in dialogs (Section 6). We also present a system that predicts different types of power (Section 7), not just hierarchical or situational power.

### 3 Data and Annotations

We use the subset of the Enron email corpus with power annotations presented in Prabhakaran et al. (2012a) for our experiments. The corpus also contains manual dialog act annotations by Hu et al. (2009), which enable us to perform the analysis of how power affects dialog behavior. The corpus contains 122 email threads with a total of 360 messages and 20,740 word tokens. There are about 8.5 participants per thread. There are 221 active participants (participants of a thread who has sent at least one email message in the thread) in the corpus. Table 1 presents the counts and percentages of active participants with each type of power in the corpus. We now define the four types of power we investigate in this paper.

**Hierarchical Power (HP):** We use the gold organizational hierarchy for Enron released by Agarwal et al. (2012) to model hierarchical power. It contains relations between 1,518 employees, and

Type of power	Count	Percentage
Hierarchical Power (HP)	18	8.1
Situational Power (SP)	81	36.7
Power over Communication (PC)	127	57.5
Influence (INFL)	11	5.0

Table 1: Annotation statistics

13,724 dominance pairs (pairs of employees such that the first dominates the second in the hierarchy, not necessarily immediately). We labeled a participant to have hierarchical power within a thread if there exist a dominance pair in the gold hierarchy such that he/she dominates any other participant in the same thread.

For the other three types of power — situational power, power over communication, and influence, we utilize the manual annotations present in the corpus of (Prabhakaran et al., 2012a).<sup>1</sup> We labeled a participant to have one of these types of power within a thread if he or she was judged to have that type of power over any other participant in the same thread. We explain the annotations in detail below with an example thread and corresponding annotations shown in Table 2; the email body contains dialog act and link annotations in [square brackets] which will be explained in Section 3.1.

**Situational Power (SP):** Person<sub>1</sub> is said to have situational power over person<sub>2</sub> if person<sub>1</sub> has power or authority to direct and/or approve person<sub>2</sub>’s actions in the current situation or while a particular task is being performed, based on the communication in the current thread. Situational power is independent of organizational hierarchy: person<sub>1</sub> with situational power may or may not be above person<sub>2</sub> in the organizational hierarchy (or there may be no organizational hierarchy at all). In our example thread, our annotator judged Kathryn to possess situational power over Leslie, Sara and Brent because Kathryn is following up on and assigning a task to others, and because Kathryn uses language that shows that she is in charge of the situation.

**Power over Communication (PC):** A person is said to have power over communication if he actively attempts to achieve the intended goals of the communication.<sup>2</sup> These are people who ask questions, request others to take action, etc., and not

<sup>1</sup>The manual annotations also capture the perception of hierarchical power. In this work, we use only the actual gold hierarchy (Agarwal et al., 2012) as described above.

<sup>2</sup>In (Prabhakaran et al., 2012a), power over communication was called “control of communication”.

From: Kathryn Cordes	
To: Leslie Hansen, Sara Shackleton, Brent Hendry	
CC: Mark Greenberg, Erik Eller, Thomas D Gros	
M1.1. Leslie Sara, and Brent: [Conventional]	
M1.2. Could I get an update on were we are with the top 20 customer amendments. [Req-Info]; [Flink1.2]	
M1.3. Last week we got 5 amendments for power physical but we still haven't received any amendments for financial. [Inform]	
M1.4. Entergy-Koch is very interested in ConfirmLogic and have asked for the amendments. [Inform]	
M1.5. When can we get the amendments for Entergy-Koch completed? [Req-Info]; [Flink1.5]	
M1.6. Thanks, [Conventional]	
M1.7. KC [Conventional]	
From: Brent Hendry	
To: Mark Taylor	
M2.1. I have just finished the draft for our internal legal review and sent it around. [Inform]	
M2.2. There are still a lot of work to be done but I do not know when everyone will have time to look at this considering how much other work there is. [Inform]	
M2.3. How should we respond considering she has copied Tom Gros? [Req-Info]; [Blink1.5]; [Flink2.3]	
Person with SP	Kathryn Cordes
Person with PC	N/A
Person with INFL	Mark Taylor
Overt Display of Power	M1.2

Table 2: Example thread with power annotations

people who simply respond to questions or perform actions when directed to do so. There could be multiple such participants in a given thread. In our example thread, no one was judged to have power over communication since the communication is broken into two separate interactions of just one message each — one from Kathryn to everyone and the other between Brent and Mark.

**Influence (INFL):** A person is defined to have influence if she 1) has credibility in the group, 2) persists in attempting to convince others, even if some disagreement occurs, 3) introduces topics/ideas that others pick up on or support, and 4) is a group participant but not necessarily active in the discussion(s) where others support/credit her. In addition, the influencer's ideas or language may be adopted by others and others may explicitly recognize influencer's authority. In our example,

our annotator judged Mark to have influence over Brent since the latter seeks advice from the former on how to deal with the situation.

### 3.1 Dialog Act Annotations

The corpus we used contains manual dialog act annotations as described in Hu et al. (2009). We use these annotations to model the dialog structure of the communication thread. For each message, Hu et al. (2009) assign a Dialog Act (DA) label to each segment of text with a coherent communicative function. The label could be one of the following: ReqAction, ReqInfo, Inform, InformOffline,<sup>3</sup> Conventional, and Commit. In addition, the segments are linked by three types of links to reflect the dialog structure. These links capture the patterns of local alternation between an initiating dialog act and a responding one. A **forward link** (Flink) is the analog of a “first pair-part” of an adjacency pair, is restricted to ReqInfo and ReqAction segments. The responses to such requests are assigned a **backward link** (Blink). If an utterance can be interpreted as a response to a preceding segment, it gets a Blink even where the preceding segment has no Flink. The preceding segment taken to be the “first pair-part” of the link is assigned a **secondary forward link** (SFlink).

### 3.2 Overt Display of Power

Our corpus also contains the **overt display of power** (ODP) (Prabhakaran et al., 2012b) annotations. An utterance is defined to have an ODP if it is interpreted as creating additional constraints on the response beyond those imposed by the general dialog act. Syntactically, an ODP can be an imperative, a question, or a declarative sentence. In our example thread, utterance M1.2 is an instance of ODP. The inter-annotator agreement value ( $\kappa$ ) of ODP annotations was 0.67.

## 4 Reliability of Annotations

The power annotations in the corpus are performed by a single annotator and capture her perception of the overall power structure among the participants of the interaction. To verify the reliability of these annotations, we performed an independent inter-annotator agreement (IAA) study on a subset of 47 threads from the corpus. We trained

<sup>3</sup>Sometimes, the Inform act refers to a previous act of communication which did not happen in the email thread itself. Such cases are marked as Offline.



two annotators — AnnA and AnnB — using the same annotation manual described in (Prabhakaran et al., 2012a) and compared the annotations they produced on the selected threads. Annotators were asked to read the entire thread before performing the annotations. They are also asked to provide, in free-form English, a short “power narrative” which describes their perception of the overall power structure among the discourse participants of that thread. Annotators build a fairly consistent mental image of a power narrative — an outline of the power structure between the participants — based on various indicators from across the thread. Their individual power annotations are based on this power narrative. Hence, the cognitive process behind labeling a participant to have a particular type of power is not a binary decision the annotator makes for each participant. However, evaluating agreement on such a formulation is not straightforward. Thus, for the purpose of this IAA study, we port this task into a binary decision task of identifying whether participant X has power of type P or not.

There were 289 participants in the selected 47 threads. The  $\kappa$  values obtained for each type of power is shown in Table 3 under Round 1. Since the  $\kappa$  values obtained in round 1 were only fair to moderate, we performed another round of training and inter annotator study. For this round, AnnB was not available, and we hired another annotator AnnC. The  $\kappa$  values obtained between AnnA and AnnC on another set of 10 threads is presented in Table 3 under Round 2.

Type of power	Round 1	Round 2
Situational Power (SP)	0.47	0.47
Power over Communication (PC)	0.27	0.76
Influence (INFL)	0.50	0.79

Table 3: Inter Rater Agreement ( $\kappa$ )

The  $\kappa$  values obtained in both round 1 and round 2 are in the range of those previously reported for similar tasks (e.g., 0.18 for managerial influence and 0.52 for establishing solidarity (Bracewell et al., 2012); 0.72 for influence (Biran et al., 2012)). The agreement in round 2 improved considerably for both PC and INFL after the second round of training. The issue of moderate agreement for SP and its possible reasons are discussed in detail in (Prabhakaran et al., 2012c). For the rest of this paper, we use the original annotations that were present in the corpus.

## 5 Dialog Behavior

We use five sets of features to capture the dialog behavior of participants: dialog act percentages (**DAP**), dialog link counts (**DLC**), positional (**PST**), verbosity (**VRB**), and overt displays of power (**ODP**). The specific features within each set are listed in Table 4. PST and VRB are readily derivable from the data, without any annotations.

Set	Features
DAP	ReqAction, ReqInform, Inform, InformOffline, Conventional, Commit
DLC	Flink, Slink, Blink, Clink, Dlink, DlinkRatio
PST	Initiator, FirstMsg, LastMsg
VRB	MsgCount, MsgRatio, TokenCount, TokenRatio, TokensPerMsg
ODP	ODPCount

Table 4: Feature Sets

DAP captures the percentages of each dialog act labels in each participant’s utterances. DLC captures the metrics on various kinds of links in each participant’s messages. Flink, Slink and Blink corresponds to counts of respective link annotations in participants’ messages. We refer to Flinks with one or more backward links as connected links (Clink) and those with no matching Blink as dangling links (Dlink). A dangling link denotes a request that was ignored. The DlinkRatio is the ratio of Dlinks to Flinks for a participant. This captures what percentage of a participant’s requests went unanswered. PST captures the positions within the thread where the participant joined and left the conversation. Initiator is a binary feature capturing whether the participant initiated the thread or not. FirstMsg and LastMsg are real valued features between 0 and 1, capturing the relative position of the first and last messages by the participant. VRB features are self explanatory. ODP captures the number of instances of ODP in the messages by each participant.

## 6 Variations in Manifestations of Power

In this section, we present the results of a statistical analysis of the dialog features with respect to people with the four types of power. For each type of power (HP, SP, INFL and PC), we consider two populations of people who participated in the dialog:  $\mathcal{P}$ , those judged to have that type of power, and  $\mathcal{N}$ , those not judged to have that power. Then, for each feature, we perform a two-sample, two-tailed t-test comparing means of feature values of

Set	Features	HP	SP	PC	INFL
DAP	ReqAction	0.10 0.02 <sub>0.23</sub>	<b>0.07 0.01</b> <sub>0.01</sub>	0.03 0.04 <sub>0.48</sub>	<b>0.0 0.04</b> <sub>6.9E-5</sub>
	ReqInform	0.10 0.11 <sub>0.87</sub>	0.10 0.12 <sub>0.70</sub>	0.11 0.11 <sub>0.91</sub>	0.09 0.11 <sub>0.73</sub>
	Inform	0.56 0.60 <sub>0.63</sub>	0.56 0.63 <sub>0.10</sub>	0.60 0.61 <sub>0.79</sub>	<b>0.78 0.59</b> <sub>0.01</sub>
	InformOffline	<b>0.00 0.005</b> <sub>0.04</sub>	0.003 0.005 <sub>0.62</sub>	<b>0.008 0.0</b> <sub>0.04</sub>	<b>0.0 0.005</b> <sub>0.04</sub>
	Conventional	0.23 0.24 <sub>0.96</sub>	0.25 0.23 <sub>0.35</sub>	0.24 0.23 <sub>0.81</sub>	<b>0.13 0.24</b> <sub>0.04</sub>
	Commit	0.0 0.002 <sub>0.21</sub>	0.001 0.003 <sub>0.51</sub>	0.001 0.004 <sub>0.44</sub>	0.0 0.002 <sub>0.21</sub>
DLC	Flink	0.56 0.74 <sub>0.27</sub>	<b>0.98 0.59</b> <sub>0.03</sub>	<b>0.91 0.49</b> <sub>6.2E-3</sub>	0.45 0.74 <sub>0.35</sub>
	SFlink	0.16 0.34 <sub>0.09</sub>	<b>0.49 0.24</b> <sub>0.02</sub>	<b>0.43 0.21</b> <sub>0.01</sub>	0.64 0.32 <sub>0.07</sub>
	Blink	0.94 0.61 <sub>0.23</sub>	0.72 0.59 <sub>0.40</sub>	<b>0.41 0.94</b> <sub>1.7E-4</sub>	1.00 0.61 <sub>0.39</sub>
	Clink	<b>0.27 0.61</b> <sub>0.04</sub>	<b>0.83 0.44</b> <sub>7.1E-3</sub>	<b>0.75 0.35</b> <sub>6.9E-4</sub>	0.73 0.57 <sub>0.46</sub>
	Dlink	0.44 0.49 <sub>0.79</sub>	0.64 0.39 <sub>0.08</sub>	0.58 0.35 <sub>0.06</sub>	0.36 0.49 <sub>0.67</sub>
	DlinkRatio	0.39 0.24 <sub>0.24</sub>	<b>0.33 0.21</b> <sub>0.05</sub>	0.27 0.24 <sub>0.57</sub>	0.18 0.26 <sub>0.55</sub>
PST	Initiator	<b>0.27 0.57</b> <sub>0.02</sub>	<b>0.68 0.48</b> <sub>3.3E-3</sub>	<b>0.88 0.11</b> <sub>3.4E-44</sub>	0.64 0.55 <sub>0.58</sub>
	FirstMsg	<b>0.34 0.19</b> <sub>0.02</sub>	<b>0.13 0.24</b> <sub>1.1E-3</sub>	<b>0.05 0.40</b> <sub>1.4E-28</sub>	0.16 0.21 <sub>0.55</sub>
	LastMsg	0.47 0.37 <sub>0.08</sub>	0.41 0.36 <sub>0.21</sub>	<b>0.31 0.47</b> <sub>1.9E-5</sub>	0.32 0.38 <sub>0.51</sub>
VRB	MsgCount	1.33 1.46 <sub>0.47</sub>	<b>1.68 1.32</b> <sub>0.03</sub>	<b>1.62 1.22</b> <sub>1.3E-3</sub>	1.45 1.45 <sub>0.99</sub>
	MsgRatio	0.48 0.52 <sub>0.47</sub>	0.54 0.50 <sub>0.18</sub>	<b>0.61 0.39</b> <sub>2.8E-15</sub>	0.45 0.52 <sub>0.19</sub>
	TokenCount	53.22 91.53 <sub>0.06</sub>	<b>113.04 74.19</b> <sub>0.02</sub>	<b>121.38 43.90</b> <sub>1.1E-8</sub>	143.55 85.54 <sub>0.10</sub>
	TokenRatio	<b>0.35 0.54</b> <sub>0.04</sub>	<b>0.62 0.47</b> <sub>2.1E-3</sub>	<b>0.72 0.26</b> <sub>1.0E-28</sub>	0.63 0.52 <sub>0.26</sub>
	TokensPerMsg	39.73 63.45 <sub>0.13</sub>	73.22 54.76 <sub>0.07</sub>	<b>78.27 38.91</b> <sub>1.3E-5</sub>	118.94 58.52 <sub>0.09</sub>
ODP	ODPCount	0.50 0.36 <sub>0.30</sub>	<b>0.78 0.14</b> <sub>6.0E-8</sub>	<b>0.49 0.21</b> <sub>2.6E-3</sub>	<b>0.09 0.39</b> <sub>0.01</sub>

Table 5: Variations in manifestations of power on feature values:  $\text{mean}(\mathcal{P}) | \text{mean}(\mathcal{N})_{p\text{-value}}$   
 $\mathcal{P}$ : people judged to have power;  $\mathcal{N}$ : people judged not to have power; Values with  $p \leq 0.05$  are boldfaced  
Types of power - SP: Situational power, HP: Hierarchical power, PC: Power over Communication, INFL: Influence;  
Features - DAP: Dialog acts, DLC: Dialog links, PST: Positional, VRB: Verbosity, ODP: Overt display of power

$\mathcal{P}$  and  $\mathcal{N}$ . Table 5 presents means of each feature value for both populations  $\mathcal{P}$  and  $\mathcal{N}$  (as “ $\text{mean}(\mathcal{P}) | \text{mean}(\mathcal{N})$ ”) along with the p-value associated with the t-test as the subscript. For  $p < 0.05$ , we reject the null hypothesis and consider the feature to be statistically significant (boldfaced in Table 5).

We find many features which are statistically significant, which suggests that power types are reflected in the dialog structure. The t-test results also show that significance of features differ considerably from one type of power to another, which suggests that different power types are reflected differently in the dialog structure, and that they are thus indeed different types of power.

For HP, we find that people with HP are less active in threads than those without. For example, persons with hierarchical power tend to talk less within a thread (TokenRatio). They tend to start participating much later in the threads (FirstMsg) and do not initiate threads often (Initiator). SP and PC manifest in stark contrast from HP. Persons with SP and persons with PC both tend to talk more within a thread (TokenRatio). They also tend to be the initiators of the thread (Initiator) or start participating in the thread closer to the beginning (FirstMsg). SP and PC have many other features which are also statistically significant. For

example, they send significantly more messages (MsgCount). They also have significantly more instances of overt displays of power (ODPCount) than others. It is interesting to note that ODPCount was not a significant feature for HP. It suggests that bosses don’t always display their power overtly when they interact. SP and PC also differ from one another. For example, those with SP tend to request actions (ReqAction) significantly more than those without. However, this was not significant in case of PC. Similarly, the number of back links (Blink) was not a significant feature for SP. But, people with PC tend to have significantly fewer back links (Blink) than those without.

This finding — people with PC have fewer back links — is interesting, since it aligns PC with the characterization of control by Walker and Whittaker (1990). According to them, control over a discourse segment is determined by whether the participant provide unsolicited information in the dialog or not. In the dialog act annotation scheme we use, solicited information (in other words, responses to requests and commands) places an obligatory Blink on the corresponding text segment. Hence, the fact that people with PC have significantly larger contributions to the dialog (VRB features), but with fewer back

links, suggest that most of their contribution is unsolicited information. This is in line with Walker and Whittaker (1990)’s definition of control over discourse segments.

Although INFL has fewer data points, we found a few significant features for INFL. People with INFL never request actions (ReqAction) as opposed to those with SP who request actions more frequently than others. Also, people with INFL tend to have significantly more inform utterances (Inform). They also have significantly fewer overt displays of power (ODPCount) than others, a stark contrast to those with SP and PC.

The statistical measures presented in previous section are exploratory in nature, presenting tests on all combinations of features and power types. We do not draw theoretical conclusions from the specific combination of interactions that are found statistically significant. Hence, we did not apply any corrections for multiple tests in statistical significance for individual features. When we apply, the Bonferroni correction for multiple tests to adjust the p-value for number of test performed (threshold =  $0.05/84 = 6.0E-4$ ), 10 features would still remain statistically significant. Hence the global null hypothesis that the features we considered do not interact with the power types would still be rejected.

## 7 Predicting Persons with Power

In this section, we present a system to predict whether a person has a given type of power in the context of an email thread. We show that different sets of features are helpful to detect different types of power. We build a separate binary classifier for each power type predicting whether or not a given participant in a communication thread has that type of power or not. Since our dataset is skewed especially for HP & INFL (with very few persons with power), we balanced our dataset by up-sampling minority class instances in the training step. This has proven useful in cases of unbalanced datasets (Japkowicz, 2000). All results presented below have been obtained after balancing the training folds in cross validation; the test folds remain unchanged. We used the tokenizer, POS tagger, lemmatizer and SVMLight (Joachims, 1999) wrapper in the ClearTK (Ogren et al., 2008) package. The ClearTK wrapper for SVMLight internally shifts the prediction threshold based on a posterior probabilistic score calcu-

Type	Feature set	P	R	F
HP	Random	16.6	38.9	11.3
	AlwaysTrue	8.1	100.0	15.0
	LEX	0.0	0.0	0.0
	VRB	<b>16.7</b>	44.4	24.2
	PST	13.8	<b>72.2</b>	23.2
	DAP	16.0	22.2	18.6
	DLC	15.3	61.1	<b>24.4</b>
SP	ODP	15.3	50.0	23.4
	VRB+PST+ODP	20.9	50.0	29.5
	Random	36.7	49.4	42.1
	AlwaysTrue	36.7	100.0	53.6
	LEX	54.9	55.6	55.2
	VRB	43.9	70.4	54.0
	PST	45.1	67.9	54.2
PC	DAP	40.9	75.3	53.0
	DLC	49.6	<b>75.3</b>	59.8
	ODP	<b>71.2</b>	51.9	<b>60.0</b>
	DLC+ODP	59.4	70.4	64.4
	Random	57.5	51.2	54.2
	AlwaysTrue	57.5	100.0	73.0
	LEX	70.2	78.0	73.9
INFL	VRB	78.7	84.3	81.4
	PST	<b>91.8</b>	88.2	<b>90.0</b>
	DAP	60.5	<b>92.9</b>	73.3
	DLC	74.3	81.9	77.9
	ODP	74.6	34.7	47.3
	PST	91.8	88.2	90.0
	Random	5.2	54.6	9.5
INFL	AlwaysTrue	5.0	100.0	9.5
	LEX	0.0	0.0	0.0
	VRB	8.1	81.8	14.8
	PST	4.6	45.5	8.4
	DAP	6.9	63.6	12.4
	DLC	<b>13.7</b>	63.6	<b>22.6</b>
	ODP	6.2	<b>90.9</b>	11.6
DLC	13.7	63.6	22.6	

Table 6: Cross validation results

SP: Situational power, HP: Hierarchical power  
PC: Power over communication, INFL: Influence  
VRB: Verbosity, PST: Positional, DAP: Dialog acts, DLC: Dialog links, LEX: Lexical, ODP: Overt display of power

lated using Lin et al. (2007)’s algorithm.

We first find the best performing subset of features for each feature set by exhaustive search within the set. Once we have the best subset of each feature set, we do another round of exhaustive search combining best performers of each set to find the overall best performing feature subset. We report micro-averaged (P)recision, (R)ecall and (F)-measure on 5-fold cross validation for each power type. We experimented with a linear kernel and a quadratic kernel; the latter performed better. All results presented in this paper are obtained using a quadratic kernel.

Table 6 shows cross validation results for all

four types of power for each set of features.<sup>4</sup> The corpus was split into folds at the thread level. We present two simple baseline measures - **Random** and **AlwaysTrue** and a language-based baseline, **LEX**. In the Random baseline, we predict an active participant to have the particular type of power at random. In AlwaysTrue baseline, we always predict an active participant to have power. For LEX, we use only lexical features (unigrams and bigrams) from messages sent by each participant to train the SVM model described above. For each power type, the table also lists (in the last row) the best performing feature subset combination and corresponding results.

HP is hard to predict, which could partly be due to the very small number of positive training examples in the corpus. For the LEX baseline using purely word ngrams, the system did not get any correct predictions. All feature subsets outperformed the other baselines of 11.3% and 15.0% (for Random and AlwaysTrue respectively), and a combination of VRB, PST and ODP gave the best model obtaining an F measure of 29.5%.

For SP, the best performing individual feature sets are ODP and DLC, both at or near 60.0%. While ODP gave a high precision (71.2%) model, DLC gave a high recall (75.3%) model, the combination of both gave the best performing system with an F measure of 64.4%.

For PC, the best single feature was FirstMsg (relative position of first message). This is because the person with the power over communication is almost always the initiator of the thread. Note that the notion of PC is not defined in terms of positional features: annotators were asked to find the participants who “actively attempt to achieve the intended goals of the communication”. It is our finding that those who are in PC were also the ones who did initiate the thread. It is also worth noting that ODP is the worst performer for PC which is in contrast with the case of SP, supporting the claim that these two types of power are in fact different.

INFL is another very hard class to predict, again, possibly partly due to the very small number of positive training examples. The simple baseline F measures were both 9.5, while the LEX did not produce any correct predictions at all. All feature sets except PST outperformed these baseline measures. The best performance was obtained

by DLC with counts of Blinks, Flinks, Dlinks and SFlinks as features.

For assessing statistical significance of F measure improvements over baseline, we used the Approximate Randomness Test (Yeh, 2000). We found the improvements to be statistically significant for SP ( $p = 0.001$ ), HP ( $p=0.001$ ) and PC ( $p = 0.01$ ) with a threshold for significance at  $p = 0.05$ . However, for INFL, the improvement was not statistically significant ( $p = 0.3$ ). The statistical significance of SP, HP and PC would hold even after applying Bonferroni correction for multiple tests.

## 8 Conclusion and Future Work

We studied four types of power between participants of written dialog. We have shown that these types of power are manifested very differently with respect to the features we are using, which validates our claim that these are indeed different types of power. We also presented a supervised learning system to predict persons with one of the types of power in written dialog yielding encouraging results. We have shown that dialog features are very significant in predicting power relations in online written communication.

In future work, we intend to try predicting power relations between pairs of participants. It would be interesting to see how dialog features correlate with the other direction of power; that is from a submitter to an exerciser of power. We will investigate the use of additional features related to the dialog participants, such as gender. We will also investigate using a dialog act tagger, link predictor and an ODP tagger to build a fully automatic power predicting system. We would also like to extend this work to other genres of written communication like discussion forums and blogs.

## 9 Acknowledgments

This paper is based upon work supported by the Johns Hopkins Human Language Technology Center of Excellence and by the DARPA DEFT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We also thank several anonymous reviewers, including three SIGDIAL reviewers, for their constructive feedback.

---

<sup>4</sup>Results for SP were presented in (Prabhakaran et al., 2012c). We present them here for comparison.

## References

- A. Agarwal, A. Omuya, A. Harnly, and O. Rambow. 2012. A Comprehensive Gold Standard for the Enron Organizational Hierarchy. In *Proceedings of the 50th Annual Meeting of the ACL (Vol. 2: Short Papers)*, pages 161–165, Jeju Island, Korea, July. ACL.
- O. Biran, S. Rosenthal, J. Andreas, K. McKeown, and O. Rambow. 2012. Detecting influencers in written online conversations. In *Proceedings of the Second Workshop on Language in Social Media*, pages 37–45, Montréal, Canada, June. ACL.
- D. B. Bracewell, M. Tomlinson, and H. Wang. 2012. A motif approach for identifying pursuits of power in social discourse. In *ICSC*, pages 1–8. IEEE Computer Society.
- P. Bramsen, M. Escobar-Molano, A. Patel, and R. Alonso. 2011. Extracting social power relationships from natural language. In *The 49th Annual Meeting of the ACL*, pages 773–782. ACL.
- C. Danescu-Niculescu-Mizil, L. Lee, B. Pang, and J. Kleinberg. 2012. Echoes of power: language effects and power differences in social interaction. In *Proceedings of the 21st international conference on WWW*, New York, NY, USA. ACM.
- J. R. French and B. Raven. 1959. The Bases of Social Power. In Dorwin Cartwright, editor, *Studies in Social Power*, pages 150–167. Univ. of Mich. Press.
- E. Gilbert. 2012. Phrases that signal workplace hierarchy. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12*, pages 1037–1046, New York, NY, USA. ACM.
- J. Hu, R. Passonneau, and O. Rambow. 2009. Contrasting the Interaction Structure of an Email and a Telephone Corpus: A Machine Learning Approach to Annotation of Dialogue Function Units. In *Proceedings of the SIGDIAL 2009 Conference*, London, UK, September. ACL.
- N. Japkowicz. 2000. Learning from imbalanced data sets: Comparison of various strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*.
- T. Joachims. 1999. Making Large-Scale SVM Learning Practical. In B. Schölkopf, C. J.C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, USA. MIT Press.
- P. W. Jordan and B. Di Eugenio. 1997. Control and initiative in collaborative problem solving dialogues. In *Working Notes of the AAAI Spring Symposium on Comp. Models for Mixed Initiative*, pages 81–84.
- H. Lin, C. Lin, and R. C. Weng. 2007. A Note on Platt’s Probabilistic Outputs for Support Vector Machines. *Mach. Learn.*, 68:267–276, October.
- M. A. Locher. 2004. *Power and politeness in action: disagreements in oral communication*. Language, power, and social process. M. de Gruyter.
- S. H. Ng, D. Bell, and M. Brooke. 1993. Gaining turns and achieving high in influence ranking in small conversational groups. *British Journal of Social Psychology*, pages 32, 265–275.
- S. H. Ng, M Brooke, , and M. Dunne. 1995. Interruption and in influence in discussion groups. *Journal of Language and Social Psychology*, pages 14(4),369–381.
- W. M. O’Barr. 1982. *Linguistic evidence: language, power, and strategy in the courtroom*. Studies on law and social control. Academic Press.
- P. V. Ogren, P. G. Wetzler, and S. Bethard. 2008. ClearTK: A UIMA toolkit for statistical natural language processing. In *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC)*.
- V. Prabhakaran, O. Rambow, and M. Diab. 2012a. Annotations for power relations on email threads. In *Proceedings of the Eighth conference on LREC*, Istanbul, Turkey, May. ELRA.
- V. Prabhakaran, O. Rambow, and M. Diab. 2012b. Predicting Overt Display of Power in Written Dialogs. In *Proceedings of the HLT-NAACL*, Montreal, Canada, June. ACL.
- V. Prabhakaran, O. Rambow, and M. Diab. 2012c. Who’s (Really) the Boss? Perception of Situational Power in Written Interactions. In *Proceedings of the 24th International Conference on COLING*, Mumbai, India. ACL.
- S. A. Reid and S. H. Ng. 2000. Conversation as a resource for in influence: evidence for prototypical arguments and social identification processes. *European Journal of Social Psych.*, pages 30, 83–100.
- R. Rowe, G. Creamer, S. Hershkop, and S.J. Stolfo. 2007. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web Mining and Social Network Anal.* ACM.
- M. Walker and S. Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th annual meeting on ACL*, pages 70–78. ACL.
- T. E. Wartenberg. 1990. *The forms of power: from domination to transformation*. Temple Univ. Press.
- A. Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on COLING - Volume 2*, pages 947–953, Stroudsburg, PA, USA. ACL.

# Evaluation of the *Scusi?* Spoken Language Interpretation System – A Case Study

Thomas Kleinbauer, Ingrid Zukerman and Su Nam Kim

Faculty of Information Technology, Monash University  
Clayton, Victoria 3800, Australia

## Abstract

We present a performance evaluation framework for Spoken Language Understanding (SLU) modules, focusing on three elements: (1) characterization of spoken utterances, (2) experimental design, and (3) quantitative evaluation metrics. We then describe the application of our framework to *Scusi?*— our SLU system that focuses on referring expressions.

## 1 Introduction

We present a performance evaluation framework for Spoken Language Understanding (SLU) modules, and describe its application to the evaluation of *Scusi?* — an SLU system that focuses on the interpretation of descriptions of household objects (Zukerman et al., 2008). Our contributions pertain to (1) the characterization of spoken utterances, (2) experimental design, and (3) quantitative evaluation metrics for an N-best list.

**Characterization of spoken utterances.** According to (Jokinen and McTear, 2010), “in diagnostic-type evaluations, a representative test suite is used so as to produce a system’s performance profile with respect to a taxonomy of possible inputs”. In addition, one of the typical aims of an evaluation is to identify components that can be improved (Paek, 2001). These two factors in combination motivate a characterization of input utterances along two dimensions: *accuracy* and *knowledge* (Section 4).

- **Accuracy** indicates whether an utterance describes an intended object precisely and unambiguously. For instance, when intending a blue plate, “the blue plate” is an accurate description if there is only one such plate in the room, while “the *green* plate” is inaccurate.
- **Knowledge** indicates how much the SLU module knows about different factors of the interpretation process, e.g., vocabulary or geometric

relations. For instance, “CPU” in “the *CPU* under the desk”<sup>1</sup> is *Out of Vocabulary (OOV)* for *Scusi?*, and the “of” in “the picture *of* a face”<sup>\*</sup> is an unknown relation.

The frequency of different values for these dimensions influence the requirements from an SLU system, and the components that necessitate additional resources, e.g., vocabulary extension.

**Experimental design.** It is generally accepted that an SLU system should exhibit reasonable behaviour by human standards. At present, in experiments that evaluate an SLU system’s performance, people speak to the system, and the accuracy of the system’s interpretation is assessed. However, this mode of evaluation, which we call *Generative*, does not address whether a system’s interpretations are plausible (even if they are wrong). Thus, in addition to a *Generative* experiment, we offer an *Interpretive* experiment. Both experiments are briefly described below. Their implementation in our SLU system is described in Section 5.

- In the *Interpretive* experiment, trial subjects and the SLU system are addressees, and are given utterances generated by a third party. The SLU system’s confidence in its interpretations is then compared with the preferences of the participants.
- In the *Generative* experiment, trial subjects are speakers, generating free-form utterances, and the SLU module and expert annotators are addressees. Gold standard interpretations for these descriptions are produced by annotators on the basis of their understanding of what was said, e.g., an ambiguous utterance has more than one correct interpretation. The SLU system’s performance is evaluated on the basis of the rank of the correct interpretations.

<sup>1</sup>Examples from our trials are marked with asterisks (\*).

These two experiments, in combination with our characterization of spoken utterances, enable the comparison of system and human interpretations under different conditions.

**Quantitative evaluation metrics.** Automatic Speech Recognizers (ASRs) and parsers often return N-best hypotheses to SLU modules, while many SLU systems return only one interpretation (DeVault et al., 2009; Jokinen and McTear, 2010; Black et al., 2011). However, maintaining N-best interpretations at the semantic and pragmatic level enables a Dialogue Manager (DM) to examine more than one interpretation, and discover features that guide appropriate responses and support error recovery. This ranking requirement, together with our experimental design, motivates the following metrics (Section 6).

- For *Interpretive* experiments, we propose correlation measures, such as Spearman rank or Pearson correlation coefficient, to compare participants' ratings of candidate interpretations with the scores given by an SLU system.
- For *Generative* experiments, we provide a broad view of an SLU system's performance by counting the utterances that it *CantRepresent*, and among the remaining utterances, counting those for which a correct interpretation was *NotFound*. We obtain a finer-grained view using fractional variants of the Information Retrieval (IR) metrics *Recall* (Salton and McGill, 1983) and *Normalized Discounted Cumulative Gain (NDCG)* (Järvelin and Kekäläinen, 2002), which handle equiprobable interpretations in an N-best list. We also compute @*K* versions of these metrics to represent the relation between rank and performance.

In the next section, we discuss related work, and in Section 3, we outline our system *Scusi?*. In Section 4, we present our characterization of descriptions, followed by our experimental design and evaluation metrics. The results obtained by applying our framework to *Scusi?* are described in Section 7, followed by concluding remarks.

## 2 Related Work

As mentioned above, our contributions pertain to the characterization of spoken utterances, experimental design, and quantitative metrics.

**Characterization of spoken utterances.** Most evaluations of SLU systems characterize input

utterances in terms of ASR *Word Error Rate (WER)*, e.g., (Hirschman, 1998; Black et al., 2011). Möller (2008) provides a comprehensive collection of interaction parameters for evaluating telephone-based spoken dialogue services, which pertain to different aspects of an interaction, viz communication, cooperativity, task success, and spoken input. Our characterization of spoken utterances along the accuracy and knowledge dimensions is related to Möller's task success category. However, in our case, these features pertain to the context, rather than the task. In addition, our characterization is linked to system development effort, i.e., how much effort should be invested to address utterances with certain characteristics; and to evaluation metrics, in the sense that the assessment of an interpretation depends on the accuracy of an utterance, and takes into account the capabilities of an SLU system.

**Experimental design.** Evaluations performed to date are based on Generative experiments (Hirschman, 1998; Gandrabur et al., 2006; Thomson et al., 2008; DeVault et al., 2009; Black et al., 2011), which focus on correct or partially correct responses. They do not consider human interpretations for utterances with diverse characteristics, as done in our Interpretive trials.

**Quantitative evaluation metrics.** Most SLU system evaluations use IR-based metrics, such as recall, precision and accuracy, to compare the components of one interpretation of a perfect request to the components of a reference interpretation (Hirschman, 1998; Möller, 2008; DeVault et al., 2009; Jokinen and McTear, 2010). In contrast, we consider the rank of completely correct interpretations of perfect requests and partially correct interpretations of imperfect requests in an N-best list. Thomson *et al.* (2008) analyzed metrics for N-best lists, such as *Receiver Operator Characteristic*, *Weighted Semantic Error Rate* and *Normalized Cross Entropy* (Gandrabur et al., 2006); and offered the *Item Level Cross Entropy (ICE)* metric, which combines the confidence score and correctness of each of N-best interpretations. In this paper, we adapt IR-based metrics to handle equiprobable interpretations in an N-best list, and offer the *CantRepresent* and *NotFound* metrics to give a broad view of system performance. In the future, we intend to incorporate confidence/accuracy metrics, such ICE.

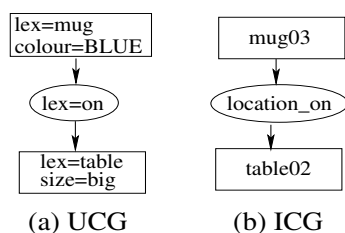


Figure 1: Sample UCG and ICG for “the blue mug on the large table”.

### 3 The *Scusi?* System

*Scusi?* is a system that implements an anytime, probabilistic mechanism for the interpretation of spoken utterances, focusing on a household context. It has four processing stages, where each stage produces multiple outputs for a given input, early processing stages may be probabilistically revisited, and only the most promising options at each stage are explored further.

The system takes as input a speech signal, and uses an ASR (Microsoft Speech SDK 6.1) to produce candidate texts. Each text is assigned a probability given the speech wave. The second stage applies Charniak’s probabilistic parser ([bllip.cs.brown.edu/resources.shtml#software](http://bllip.cs.brown.edu/resources.shtml#software)) to syntactically analyze the texts in order of their probability, yielding at most 50 different parse trees per text. The third stage applies mapping rules to the parse trees to generate *Uninstantiated Concept Graphs (UCGs)* that represent the semantics of the utterance (Sowa, 1984). The final stage produces *Instantiated Concept Graphs (ICGs)* that match the concepts and relations in a UCG with objects and relations within the current context (e.g., a room), and estimates how well each instantiation matches its “parent” UCG and the context. For example, Figure 1(a) shows one of the UCGs returned for the description “the blue mug on the large table”, and Figure 1(b) displays one of the ICGs generated for this UCG. Note that the concepts in the UCG have generic names, e.g., *mug*, while the ICG contains specific objects, e.g., *mug03* or *cup01*, which are offered as candidate matches for *lex=mug*, *color=blue*.

#### 3.1 *Scusi?*’s capabilities

*Scusi?* aims to understand requests for actions involving physical objects (Zukerman et al., 2008). Focusing on object descriptions, *Scusi?* has a vocabulary of lexical items pertaining to objects, colours, sizes and positions. For object names, this vocabulary is expanded with synonyms and near

synonyms obtained from WordNet (Fellbaum, 1998) and word similarity metrics from (Leacock and Chodorow, 1998). However, this vocabulary is not imposed on the ASR, as we do not want *Scusi?* to hear only what it wants to hear. In addition, *Scusi?* was designed to understand the colour and size of objects; the topological positional relations *on*, *in*, *near* and *at*, optionally combined with *center*, *corner*, *edge* and *end*, e.g., “the mug *near the center* of the table”; and the projective positional relations *in front of*, *behind*, *to the left/right*, *above* and *under* (topological and projective relations are discussed in detail in (Coventry and Garrod, 2004; Kelleher and Costello, 2008)). By “understanding a description” we mean mapping attributes and positions to values in the physical world. For instance, the CIE colour metric (CIE, 1995) is employed to understand colours, Gaussian functions are used to represent sizes of things compared to the size of an average exemplar, and spatial geometry is used to understand positional relations.

At present, *Scusi?* does not understand (1) *OOV* words, e.g., “the *opposite* wall”\*; (2) more than one meaning of polysemous positional relations, e.g., “*to the left of* the table”\* as “*to the left and on* the table” as well as “*to the left and next to* the table”; (3) positional relations that are complex, e.g., “*in the left near corner of* the table”\*, or don’t have a landmark, e.g., “the ball *in the center*”\*; and (4) descriptive prepositional phrases starting with “of” or “with”, e.g., “the picture *of* the face”\* and “the plant *with* the leaves”\*. However, contextual information sometimes enables the system to overcome *OOV* words. For example, *Scusi?* may return the correct ICG for “the *round* blue plate on the table” at a good rank.

Clearly, these problems can be solved by programming additional capabilities into our system. However, people will always say things that an SLU system cannot understand. Our evaluation framework can help distinguish between situations in which it is worth investing additional development effort, and situations for which other coping mechanisms should be developed, e.g., asking a clarification question or ignoring the unknown portions of an utterance (while being aware of the impact of this action on comprehension).

#### 3.2 ASR capabilities

The WER of the ASR used by *Scusi?* is 30% when trained on an open vocabulary in combination with a small language model for our corpus.



This WER is consistent with the WER obtained in the 2010 Spoken Dialogue Challenge (Black et al., 2011). In addition to the obvious problem of mis-recognized entities or actions, which yield OOV words, ASR errors often produce ungrammatical sentences that cannot be successfully parsed. For instance, one of the alternatives produced by the ASR for “the blue plate at the front of the table”<sup>\*</sup> is “*to build played* at the front *door* the table”. Further, disfluencies are often mis-heard by the ASR or cause it to return broken sentences.

#### 4 Characterization of Spoken Utterances

When describing an object or action, speakers may employ a wrong lexical item, or use a wrong attribute. For instance, “the green *couch*”<sup>\*</sup> was described when intending a green bookcase. In addition, when describing objects, speakers may under-specify them, e.g., ask for “the pink mug” when there are several such mugs; provide inconsistent specifications that do not match any object perfectly, yielding no candidates or several partial candidates, e.g., request “the large blue mug” when there is a large pink mug and a small blue mug; omit a landmark, e.g., “the ball in the center”<sup>\*</sup>; or employ words or constructs unknown to an SLU module, e.g., “the *exact* center”<sup>\*</sup>.<sup>2</sup> These situations, which affect the performance of an SLU system, are characterized along the following two dimensions: *accuracy* and *knowledge*.

- **Accuracy** – We distinguish between *Perfect* and *Imperfect* utterances. An utterance is perfect if it matches at least one object or action in the current context in every respect. In this case, an SLU module should produce one or more interpretations that match perfectly the utterance. If every object or action in the context mismatches an utterance at least in one aspect, the utterance is *imperfect*. In this case, we consider *reasonable* interpretations (that match the request well but not perfectly) to be the Gold standard. The **number** of Gold interpretations is an attribute of accuracy: an utterance may match (perfectly or imperfectly) 0, 1 or more than 1 interpretation.
- **Knowledge** – If all the words and syntactic constructs in an utterance are understood by an SLU module (Section 3.1), the utterance is deemed *known*, otherwise, it is *unknown*.

<sup>2</sup>People often over-specify their descriptions, e.g., “the large red mug” when there is only one red mug (Dale and Reiter, 1995). Such over-specifications are not problematic.

To illustrate these concepts, a description that contains only known words, and matches two objects in the context in every respect, is classified as *known-perfect*>1.

### 5 Experimental Design

We devised two experiments to assess an SLU system’s performance: *Interpretive*, where the participants and the SLU system are the addressees (Section 5.1), and *Generative*, where the participants are the speakers and the SLU module is the addressee (Section 5.2).

In both experiments, we evaluate the performance of an SLU system on the basis of complete interpretations of an utterance, which in *Scusi?*’s case is a description. For example, given “the pink ball near the table”, all the elements of an ICG must try to match this description and the context. That is, if `ball01` is pink, but it is *on* `table02`, the ICG `ball01-location_near-table02` will have a good description match but a bad reality match, while the opposite happens for ICG `ball01-location_on-table02`.

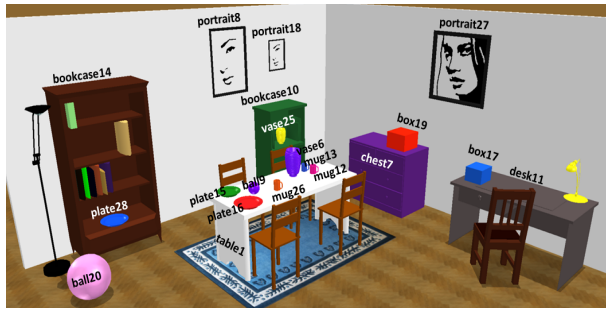
#### 5.1 Interpretive trial

This experiment tests whether *Scusi?*’s understanding matches the understanding of a relatively large population under different accuracy conditions. We focus on imperfect and ambiguous descriptions, as they pose a greater challenge to people than perfect descriptions. The trial consists of a Web-based survey where participants were given a picture of a room and 9 descriptions generated by the authors (Figure 2). For each description, participants were asked to rate each of 20 labeled objects based on how well they match the description, where a rating of 10 denotes a “perfect match” and a rating of 0 denotes “no match”.

Our Web survey was done by 47 participants, resulting in  $47 \times 20$  scores for each description. These scores were averaged across participants, yielding a single score for each labeled object for each of our 9 descriptions.

#### 5.2 Generative trial

In this experiment, trial subjects generated free-form, spoken descriptions to identify three designated objects in each of four scenarios. The scenarios, which were designed to test different functionalities of *Scusi?*, contain between 8 and 16 objects (Figure 3 shows two scenarios). The annotators provided the Gold standard interpretations for

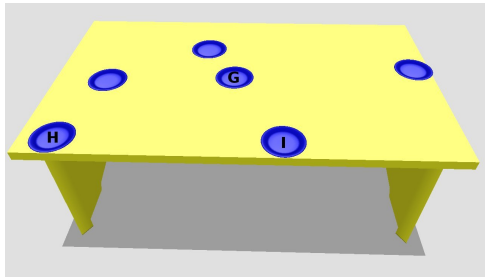


(a) Room with labeled objects

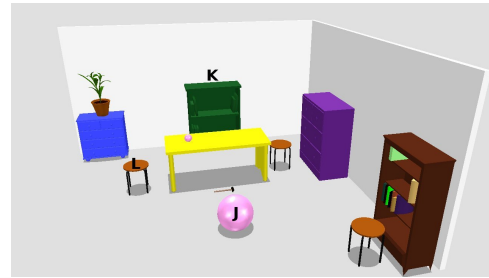
1. *the plate next to the ball* *perfect*>1
2. *the large blue box* *imperfect*>1
3. *the red dish* *perfect*=1
4. *the brown bookcase under the portrait* *imperfect*>1
5. *the orange mug near the vase* *imperfect*>1
6. *the large plate* *perfect*>1
7. *the large green bookcase near the chest* *imperfect*=1
8. *the large ball on the table* *imperfect*>1
9. *the portrait above the bookcase* *perfect*=1

(b) Descriptions with their characterization

Figure 2: Context visualization and object descriptions used in the Interpretive experiment.



(a) Projective relations and “end, edge, corner” and “center” of a table



(b) Colour, size, positional relation and intervening object in a room

Figure 3: Two of the scenarios used in the Generative experiments.

a description on the basis of what they understood (rather than using the designated referents). Each annotator handled half of the descriptions, and the other annotator verified the annotations. Disagreements were resolved by consensus.

Our study had 26 participants, who generated a total of 432 spoken descriptions (average length was 10 words, median 8, and the longest description had 21 words). We manually filtered out 32 descriptions that were broken up by the ASR due to pauses made by the speakers, and 105 descriptions that *Scusi?* *CantRepresent* (Section 6.2). Two sets of files were submitted to *Scusi?*: a set containing textual transcriptions of the remaining 295 descriptions, and a set containing textual alternatives produced by the ASR for each of these descriptions.

This experiment enables us to observe the frequencies of descriptions with different characteristics (Section 4), and determine their influence on performance, as well as the effect of ASR versus textual input. Table 3 displays the frequencies of the four accuracy classes of descriptions (*perfect* =1 and >1 and *imperfect* =1 and >1), and two knowledge classes (*known* and *unknown-OOV*) (Section 4). For instance, the top row shows that 197 descriptions are *known-perfect*=1 (Col-

umn 2), and 25 descriptions are *unknown-OOV* (Column 3). 18 *unknown-non-OOV* descriptions were omitted from Table 3. These descriptions have Gold ICGs, but contain word combinations that are not known to *Scusi?*, e.g., “on top of” and “at the front of”. Note the low frequencies of three of the *unknown-OOV* categories, and of the *imperfect*>1 classes. The latter suggests that, unlike our Interpretive trial, people rarely generate descriptions that are both ambiguous and inaccurate. Table 3 also displays the results obtained for the performance metrics *NotFound@K*, *FRecall@K* and *NDCG@K* (Section 6) for each accuracy-knowledge combination and for Text and ASR input; the results are described in Section 7.

## 6 Evaluation Metrics

We first consider the Interpretive trial followed by the Generative trial.

### 6.1 Interpretive trial

*Scusi?*’s understanding of each description was compared with that of our trial subjects by calculating the Spearman rank correlation coefficient and Pearson correlation coefficient between the average of the scores of the subjects’ ratings for each object, and the probability assigned

Table 1: Descriptions that cannot be represented.

	Positional relation			Others and Prep. Phrase “with”/“of”
	Poly- semous	Complex	No Landm.	
<i>perfect=1</i>	9	29	0	9
<i>perfect&gt;1</i>	5	15	0	4
<i>imperfect=1</i>	6	13	18	3
<i>imperfect&gt;1</i>	2	2	1	0
TOTAL	22	59	19	16

by *Scusi?* to the top-ranked correct interpretation with the corresponding head object, e.g., `plate16-near-ball09` for the first description in Figure 2(b). The results for the Spearman rank and Pearson correlation coefficient appear in Section 7.1.

## 6.2 Generative trial

We first describe our broad metrics, followed by the fine-grained metrics.

**CantRepresent** counts the number of utterances that an SLU system cannot represent, which are a subset of the *unknown* utterances, and are excluded from the rest of the evaluation. Table 1 displays the frequencies of such descriptions and their causes (11 descriptions had more than one problem). As shown in Table 1, complex positional relations, e.g., “*the left front corner*”\*, account for most of the problems.

**NotFound@K** counts the number of representable utterances for which no correct interpretation was found within rank  $K$ . **NotFound@∞** considers all the interpretations returned by an SLU system. It is worth noting that *NotFound* utterances are included when calculating the following metrics.

**Precision@K and Recall@K.** The @ $K$  versions of precision and recall evaluate performance for different cut-off ranks  $K$ .

**Precision@K** is simply the number of correct interpretations at rank  $K$  or better divided by  $K$ . **Recall@K** is defined as follows:

$$\text{Recall@}K(d) = \frac{|CF(d) \cap \{I_1, \dots, I_K\}|}{|C(d)|},$$

where  $C(d)$  is the set of correct interpretations for utterance  $d$ ,  $CF(d)$  is the set of correct interpretations found by an SLU module, and  $I_j$  denotes an interpretation with rank  $j$ .

Contrary to IR settings, where typically there are many relevant documents, in language understanding situations, there is often one correct interpretation for an utterance (Table 3). If this interpretation is ranked close to the top, **Precision@K**

will be constantly reduced as  $K$  increases. Hence, we eschew this measure when evaluating the performance of an SLU system.

An SLU module may return several equiprobable interpretations, some of which may be incorrect. The relative ranking of these interpretations is arbitrary, leading to non-deterministic values for **Recall@K** — a problem that is exacerbated when  $K$  falls within a set of such equiprobable interpretations. This motivates a variant of **Recall@K**, denoted **FRecall@K** (*Fractional Recall*), that allows us to represent the arbitrariness of the ranked order of equiprobable interpretations, as follows:

$$\text{FRecall@}K(d) = \frac{\sum_{j=1}^K fc(I_j)}{|C(d)|}, \quad (1)$$

where  $fc$  is the fraction of correct interpretations among those with the same probability as  $I_j$  (this is a proxy for the probability that  $I_j$  is correct):

$$fc(I_j) = \frac{c_j}{h_j - l_j + 1}, \quad (2)$$

where  $l_j$  is the lowest rank of all the interpretations with the same probability as  $I_j$ ,  $h_j$  the highest rank, and  $c_j$  the number of correct interpretations between rank  $l_j$  and  $h_j$  inclusively.

**Normalized Discounted Cumulative Gain (NDCG@K).** A shortcoming of **Recall@K** is that it considers the rank of an interpretation only in a coarse way (at the level of  $K$ ). A finer-grained account of rank is provided by **NDCG@K** (Järvelin and Kekäläinen, 2002), which discounts interpretations with higher (worse) ranks.

**DCG@K** allows the definition of a relevance measure for a result, and divides this measure by a logarithmic penalty that reflects the rank of the result. Using  $fc(I_j)$  as a measure of the relevance of interpretation  $I_j$ , we obtain

$$\text{DCG@}K(d) = fc(I_1) + \sum_{j=2}^K \frac{fc(I_j)}{\log_2 j}.$$

This score is normalized to the  $[0, 1]$  range by dividing it by the score of an ideal answer where  $|C(d)|$  correct interpretations are ranked in the first  $|C(d)|$  places, yielding

$$\text{NDCG@}K(d) = \frac{\text{DCG@}K(d)}{1 + \sum_{j=2}^{\min\{|C(d)|, N\}} \frac{1}{\log_2 j}}. \quad (3)$$

Note that **FRecall@K** is computed in relation to the number of correct interpretations, while **NDCG@K** considers the minimum of  $K$  and this number (Equations 1 and 3 respectively).

Table 2: Results of the Interpretive trials.

#	Survey	<i>Scusi?</i> $I_1$	<i>Scusi?</i> $I_2$	<i>Scusi?</i> $I_3$
1.	plate16	<b>plate16</b> –(near)→ <b>ball9</b>	plate15 –(near)→ ball9	plate28 –(near)→ ball20
2.	box17	<b>box17</b>	box19	carpet23
3.	plate16	mug26	<b>plate16</b>	mug12
4.	bookcase14	bookcase10 –(under)→ portrait18	bookcase10 –(under)→ portrait8	<b>bookcase14</b> –(instr_r)→ portrait8
5.	mug26	<b>mug26</b> –(near)→ <b>vase6</b>	mug12 –(near)→ vase6	mug13 –(near)→ vase6
6.	plate28	plate16/ <b>plate28</b>	<b>plate28</b> /plate16	plate15
7.	bookcase10	bookcase14 –(near)→ chest7	<b>bookcase10</b> –(near)→ <b>chest7</b>	bookcase14 –(recipient_r)→ chest7
8.	ball9	<b>ball9</b> –(on)→ <b>table1</b>	ball20 –(agent_r)→ table1	ball20 –(action_r)→ table1
9.	portrait18	<b>portrait18</b> –(above)→ <b>bookcase10</b>	portrait8 –(above)→ bookcase10	portrait27 –(instr_r)→ bookcase14

## 7 Results

We first discuss the results of our Interpretive trials followed by those of our Generative trials.

### 7.1 Interpretive Trials

Table 2 compares the results of the Web survey with *Scusi?*’s performance for the Interpretive trials. Column 2 indicates the object preferred by the trial subjects, and Columns 3-5 show the top-three interpretations preferred by *Scusi?* ( $I_1$ – $I_3$ ). Matches between the system’s output and the averaged participants’ ratings are boldfaced.

As seen in Table 2, *Scusi?*’s ratings generally match those of our participants, achieving a strong Pearson correlation of 0.77, and a weaker Spearman correlation of 0.63. This is due to the fact that implausible interpretations get a score of 0 from *Scusi?*, while some people still choose them, thus yielding different ranks for them.

*Scusi?*’s top-ranked interpretation matches our participants’ preferences in 5.5 cases, and its second-ranked interpretation in 2.5 cases (the fractions are for equiprobable interpretations). The discrepancies between *Scusi?*’s choices and those of our trial subjects are explained as follows: (desc. 3) “the red dish” – according to Leacock and Chodorow’s similarity metric (Section 3.1), a mug is more similar to a dish than a dinner plate, while our trial subjects thought otherwise; (desc. 4) “the brown bookcase under the portrait” – *Scusi?* penalizes heavily attributes that do not match reality (Zukerman et al., 2008), hence `bookcase14` is penalized, as it is not under any portrait; (desc. 6) “the large plate” – our participants perceived `plate28` to be larger than `plate16` although they are the same size, and hence equiprobable; (desc. 7) “the large green bookcase near the chest” – like description 4, `bookcase10` (which is green) is ranked second due to its low probability of being considered large.

Thus, according to this trial, *Scusi?*’s performance satisfies our original requirement for rea-

sonable behaviour and plausible mistakes, but perhaps it should be more forgiving with respect to mis-matched attributes.

### 7.2 Generative Trials

Table 3 displays the results for *NotFound@K*, *FRecall@K* and *NDCG@K* for  $K = 1, 3, 10, \infty$  for Text and ASR input, the four accuracy classes, and the *known* and *unknown-OOV* knowledge categories. There are 277 descriptions in total (instead of 295), as 18 *unknown-non-OOV* descriptions were omitted from Table 3 (Section 5.2). As mentioned in Section 5.2, the vast majority of the utterances belong to the *perfect=1* class (with *known* or *unknown-OOV* words), and to the *known perfect>1* and *imperfect=1* categories.

**ASR versus Text.** The *NotFound@1,3*, *FRecall@1,3* and *NDCG@1,3* metrics show that *Scusi?* yields at least one correct interpretation at the lowest (best) ranks for the vast majority of Text inputs (the discrepancy between *FRecall* and *NDCG* at low ranks is due to the way these measures are calculated, Section 6.2). This suggests that in the absence of ASR errors, if correct interpretations are found, the system’s confidence in its output is justified. As expected, the *NotFound* values are substantially higher, and the *FRecall* and *NDCG* values lower, for inputs obtained from the ASR (23% of the descriptions had one wrong word in the best ASR alternative, 21% had two wrong words, 12.5% had three, and 8.5% more than three). There is a substantial improvement in *FRecall* and *NDCG* as ranks increase, which shows that contextual information can alleviate some ASR errors. The improvement in these metrics for the *perfect>1* class, without affecting *NotFound*, indicates that *Scusi?* finds more correct interpretations for the same descriptions.

The ASR results compared to those of Text indicate that, unsurprisingly, speech recognition quality must be improved. This may be achieved through advances in ASR technology, or by pre-

Table 3: Description breakdown in terms of accuracy and knowledge, performance metrics and results.

	Known		Unknown-OOV	
	Text	ASR	Text	ASR
<i>perfect=1</i>	197		25	
<i>NotFound@1,3,10,∞</i>	9,4,2,1	73,60,49,31	8,8,8,3	16,13,11,9
<i>FRecall@1,3,10,∞</i>	0.95,0.98,0.99,0.99	0.61,0.69,0.75,0.84	0.47,0.68,0.68,0.88	0.24,0.45,0.54,0.64
<i>NDCG@1,3,10,∞</i>	0.95,0.98,0.98,0.98	0.61,0.69,0.71,0.73	0.47,0.64,0.64,0.68	0.24,0.40,0.44,0.46
<i>perfect&gt;1</i>	30		1	
<i>NotFound@1,3,10,∞</i>	2,2,1,1	13,12,10,9	0,0,0,0	0,0,0,0
<i>FRecall@1,3,10,∞</i>	0.40,0.82,0.88,0.97	0.22,0.48,0.62,0.70	0.50,1.00,1.00,1.00	0.50,1.00,1.00,1.00
<i>NDCG@1,3,10,∞</i>	0.84,0.84,0.85,0.87	0.47,0.48,0.53,0.55	1.00,1.00,1.00,1.00	1.00,1.00,1.00,1.00
<i>imperfect=1</i>	18		2	
<i>NotFound@1,3,10,∞</i>	1,1,1,0	8,7,7,5	0,0,0,0	1,1,1,1
<i>FRecall@1,3,10,∞</i>	0.91,0.94,0.94,1.00	0.56,0.59,0.61,0.72	0.51,0.54,0.64,1.00	0.03,0.08,0.26,0.50
<i>NDCG@1,3,10,∞</i>	0.91,0.94,0.94,0.95	0.56,0.59,0.60,0.61	0.51,0.54,0.58,0.66	0.03,0.07,0.14,0.20
<i>imperfect&gt;1</i>	3		1	
<i>NotFound@1,3,10,∞</i>	1,0,0,0	3,2,1,1	0,0,0,0	1,1,1,1
<i>FRecall@1,3,10,∞</i>	0.18,0.53,0.61,1.00	0.00,0.33,0.51,0.67	0.03,0.09,0.29,1.00	0.00,0.00,0.00,0.00
<i>NDCG@1,3,10,∞</i>	0.36,0.53,0.56,0.64	0.00,0.27,0.35,0.38	0.06,0.08,0.15,0.31	0.00,0.00,0.00,0.00

venting ASR errors (Gorniak and Roy, 2005; Sugiyura et al., 2009) or correcting them (López-Cózar and Callejas, 2008; Kim et al., 2013).

**Known versus Unknown-OOV.** *Perfect=1* is the only class with a substantial number of OOV words (25). Note the increase in *FRecall* up to rank @∞ for *known* ASR and *unknown-OOV* Text and ASR, which indicates that correct interpretations are returned at very high ranks when input words are not identified (*NDCG* increases only modestly, as it penalizes high ranks). The difference in performance between *known-perfect=1* and *unknown-OOV-perfect=1* suggests that it is worth improving *Scusi?*'s vocabulary coverage.

## 8 Conclusion

We offered a framework for the evaluation of SLU systems that comprises a characterization of spoken utterances, experimental design and evaluation metrics. We described its application to the evaluation of *Scusi?*—our SLU module for the interpretation of descriptions in a household context.

Our characterization of descriptions identifies frequently occurring cases, such as *perfect=1*, and rare cases, such as *imperfect>1*; and highlights the influence of vocabulary coverage on performance.

Our two types of experiments enable the evaluation of an SLU system's performance from two viewpoints: Interpretive trials support the comparison of an SLU module's performance with that of people as addressees, and Generative trials assess the performance of an SLU system when interpreting descriptions commonly spoken by users. The results of the Interpretive trial were encouraging, but they indicate that *Scusi?*'s "punitive" at-

titude to attributes that do not match reality, such as a bookcase not being under any portrait, may need to be moderated. However, as stated above, *imperfect>1* descriptions were rare in our Generative trials. The results of these trials show that development effort should be invested in (1) ASR accuracy (Kim et al., 2013); (2) vocabulary coverage; and (3) ability to represent complex, polysemous and no-landmark positional relations. In contrast, descriptive prepositional phrases starting with "with" or "of" may be judiciously ignored, or the referent may be disambiguated by asking a clarification question.

Our *CantRepresent* and *NotFound* evaluation metrics provide an overall view of an SLU system's performance. IR-based metrics have been used in the evaluation of SLU systems to compare an interpretation returned by an SLU module with a reference interpretation. In contrast, we employ *FRecall* and *NDCG* in the traditional IR manner, i.e., to assess the rank of correct interpretations in an N-best list. The relevance measure *fc* (Equation 2), which is applied to both metrics, enables us to handle equiprobable interpretations. However, rank-based evaluation metrics do not consider the absolute quality of an interpretation, i.e., the top-ranked interpretation might be quite bad. In the future, we propose to investigate confidence/accuracy metrics, such as ICE (Thomson et al., 2008), to address this problem.

## Acknowledgments

This research was supported in part by grants DP110100500 and DP120100103 from the Australian Research Council.

## References

- A. Black, S. Burger, A. Conkie, H. Hastie, S. Keizer, O. Lemon, N. Merigaud, G. Parent, G. Schubiner, B. Thomson, J.D. Williams, K. Yu, S. Young, and M. Eskenazi. 2011. Spoken dialog challenge 2010: Comparison of live and control test results. In *Proceedings of the 11th SIGdial Conference on Discourse and Dialogue*, pages 2–7, Portland, Oregon.
- CIE, 1995. *Industrial colour difference evaluation*. CIE 115-1995.
- K.R. Coventry and S.C. Garrod. 2004. *Saying, Seeing, and Acting: the psychological semantics of spatial prepositions*. Psychology Press.
- R. Dale and E. Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 18(2):233–263.
- D. DeVault, K. Sagae, and D. Traum. 2009. Can I finish? Learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the 10th SIGdial Conference on Discourse and Dialogue*, pages 11–20, London, United Kingdom.
- C.D. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. MIT Press.
- S. Gandrabur, G. Foster, and G. Lapalme. 2006. Confidence estimation for NLP applications. *ACM Transactions on Speech and Language Processing*, 3(3):1–29.
- P. Gorniak and D. Roy. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *ICMI'05: Proceedings of the 7th International Conference on Multimodal Interfaces*, pages 138–143, Trento, Italy.
- L. Hirschman. 1998. The evolution of evaluation: Lessons from the Message Understanding Conferences. *Computer Speech and Language*, 12:281–305.
- K. Järvelin and J. Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.
- K. Jokinen and M. McTear. 2010. *Spoken Dialogue Systems*. Morgan and Claypool.
- J.D. Kelleher and F.J. Costello. 2008. Applying computational models of spatial prepositions to visually situated dialog. *Computational Linguistics*, 35(2):271–306.
- S.N. Kim, I. Zukerman, Th. Kleinbauer, and F. Zavareh. 2013. A noisy channel approach to error correction in spoken referring expressions. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, Nagoya, Japan.
- C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–285. MIT Press.
- R. López-Cózar and Z. Callejas. 2008. ASR post-correction for spoken dialogue systems based on semantic, syntactic, lexical and contextual information. *Journal of Speech Communication*, 50(8-9):745–766.
- S. Möller. 2008. Evaluating interactions with spoken dialogue telephone services. In L. Dybkjær and W. Minker, editors, *Recent Trends in Discourse and Dialogue*, pages 69–100. Springer.
- T. Paek. 2001. Empirical methods for evaluating dialog systems. In *SIGDIAL'01 – Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–9, Aalborg, Denmark.
- G. Salton and M.J. McGill. 1983. *An Introduction to Modern Information Retrieval*. McGraw Hill, New York, New York.
- J.F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.
- K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura. 2009. Bayesian learning of confidence measure function for generation of utterances and motions in object manipulation dialogue task. In *Proceedings of Interspeech 2009*, pages 2483–2486, Brighton, United Kingdom.
- B. Thomson, K. Yu, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, and S. Young. 2008. Evaluating semantic-level confidence scores with multiple hypotheses. In *Proceedings of Interspeech 2008*, pages 1153–1156, Brisbane, Australia.
- I. Zukerman, E. Makalic, M. Niemann, and S. George. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 581–592, Hanoi, Vietnam.

# A Noisy Channel Approach to Error Correction in Spoken Referring Expressions

Su Nam Kim, Ingrid Zukerman, Thomas Kleinbauer and Farshid Zavareh  
Faculty of Information Technology, Monash University  
Clayton, Victoria 3800, Australia

## Abstract

We offer a noisy channel approach for recognizing and correcting erroneous words in referring expressions. Our mechanism handles three types of errors: it removes noisy input, inserts missing prepositions, and replaces mis-heard words (at present, they are replaced by generic words). Our mechanism was evaluated on a corpus of 295 spoken referring expressions, improving interpretation performance.

## 1 Introduction

One of the main stumbling blocks for Spoken Dialogue Systems (SDSs) is the lack of reliability of Automatic Speech Recognizers (ASRs) (Pellegrini and Trancoso, 2010). Recent research prototypes of ASRs yield Word Error Rates (WERs) between 15.6% (Pellegrini and Trancoso, 2010) and 18.7% (Sainath et al., 2011) for broadcast news. However, the commercial ASR employed in this research had a WER of 30% and a Sentence Error Rate (SER) (proportion of sentences for which no correct textual transcription was produced) of 65.3% for descriptions of household objects.

In addition to mis-recognized entities or actions, ASR errors often yield ungrammatical sentences that cannot be processed by subsequent interpretation modules of an SDS, e.g., “the blue plate” being mis-heard as “*to build played*”, and hesitations (e.g., “ah”s) being mis-heard as “and” or “on” — all of which happened in our trials.

In this paper, we offer a general framework for error detection and correction in spoken utterances that is based on the noisy channel model, and present a first-stage implementation of this framework that performs simple corrections of referring expressions. Our model is implemented as a pre-processing step for the *Scusi?* spoken language interpretation system (Zukerman et al., 2008; Zukerman et al., 2009).

Table 1: Spoken, heard and labeled descriptions.

Spoken:	the stool	<i>to</i>	the left of	the table
Heard:	the <i>storm</i>		the left of	the table
Labels:	<b>Object</b>	<b>Prep</b>	<b>Specifier</b>	<b>Landmark</b>
Spoken:	the plate		in	the microwave
Heard:	<i>to play</i>	<i>it</i>	in	the microwave
Labels:	<b>Object</b>	<b>Noise</b>	<b>Prep</b>	<b>Landmark</b>

The idea of the noisy channel model is that a message is sent through a channel that introduces errors, and the receiver endeavours to reconstruct the original message by taking into account the characteristics of the noisy channel and of the transmitted information (Ringger and Allen, 1996; Brill and Moore, 2000; Zwarts et al., 2010). The system described in this paper handles three types of errors: noise (which is removed), missing prepositions (which are inserted), and mis-heard words (which are replaced). Table 1 shows two descriptions that illustrate these errors. The first row for each description displays what was spoken, the second row displays what was heard by the ASR, and the third row shows the semantic labels assigned to each segment in the description by a shallow semantic parser (Section 3.2). Specifically, in the first example, the preposition “*to*” is missing, and the object “stool” is mis-heard as “*storm*”; and in the second example “the plate” is mis-heard as “*to play*”, and the noisy “*it*” has been inserted by the ASR.

Ideally, we would like to replace mis-heard words with phonetically similar words, e.g., use “plate” instead of “*play*”. However, at present, as a first step, we replace mis-heard words with generic options, e.g., “thing” for an object or landmark. Further, we insert the generic preposition “at” for a missing preposition. Thus, we deviate from the noisy channel approach in that we do not quite reconstruct the original message. Instead, we construct a grammatically correct version of this message that enables the generation of reasonable interpretations (rather than no interpretation or non-

sensical ones). For example, the mis-heard description “to play it in the microwave” in Table 1 is modified to “the thing in the microwave”. Clearly, this is not what the speaker said, but hopefully, this modified text, which describes an object, rather than an action, enables the identification of the intended object, e.g., a plate, or at least a small set of candidates, in light of the rest of the description.

Our mechanism was evaluated on a corpus of 295 spoken referring expressions, significantly improving the interpretation performance of the original *Scusi?* system (Section 6.3).

The rest of this paper is organized as follows. In the next section, we discuss related work. In Section 3, we outline the design of our system. Our probabilistic model is described in Section 4, followed by the noisy channel error correction procedure. In Section 6, we discuss our evaluation, and then present concluding remarks.

## 2 Related Research

This research combines three main elements: correction of ASR output, noisy channel models and shallow semantic parsing.

López-Cózar and Griol (2010) used lexical approaches to replace, insert or delete words in a textual ASR output, and syntactic approaches to modify tenses of verbs and grammatical numbers to better match grammatical expectations. However, these actions make *ad hoc* changes.

The noisy channel model has been employed for various NLP tasks, such as ASR output correction (Ringger and Allen, 1996), spelling correction (Brill and Moore, 2000), and disfluency correction (Johnson and Charniak, 2004; Zwarts et al., 2010). Our approach differs from the traditional noisy channel approach in that it uses a word-error classifier to model the noisy channel, and semantic information to model the input characteristics.

Shallow semantic parsers for SDSs have been used in (Coppola et al., 2009; Geertzen, 2009). Coppola *et al.* (2009) used FrameNet (Baker et al., 1998) to detect and filter the frames for target words, and employed a Support Vector Machine (SVM) classifier to perform semantic labeling. Geertzen (2009) used a shallow parser to detect semantic units only when a dependency parser failed to produce a parse tree. In contrast, our shallow semantic parser is part of a noisy channel model that post-processes the output of an ASR.

## 3 System Design

Our error correction procedure (Section 5) receives as input alternatives produced by an ASR, and generates modified versions of these alternatives. It employs the following modules: (1) a classifier that determines whether a word in a text produced by the ASR is correct; (2) a shallow semantic parser (SSP) that assigns semantic labels to segments in the text; and (3) a noisy channel error correction mechanism that decides which alterations should be made to the ASR output on the basis of the information provided by the other two modules. The resultant texts are given as input to the *Scusi?* spoken language interpretation system.

In this section, we describe the word error classifier and SSP together with our semantic labels, and report on their performance. We also provide a brief outline of the *Scusi?* system.

The performance of the classifier and SSP was evaluated in terms of accuracy over the corpus constructed to evaluate the *Scusi?* system (Kleinbauer et al., 2013). This corpus comprises 400 spoken descriptions generated by 26 speakers. We performed 13-fold cross-validation, where each fold contains two speakers (Section 6.1).

### 3.1 Word error classifier

We investigated three classifiers to determine whether a word in the ASR textual output is correct: the Weka implementation of Decision Trees (Quinlan, 1993) and Naïve Bayes classifiers (Domingos and Pazzani, 1997) ([cs.waikato.ac.nz/ml/weka/](http://cs.waikato.ac.nz/ml/weka/)), and the Mallet implementation of the linear chain Conditional Random Fields (CRF) algorithm (Lafferty et al., 2001) ([mallet.cs.umass.edu](http://mallet.cs.umass.edu)).

The best performance was obtained by the Decision Tree, which yielded an average accuracy of 80.9% over the 13 folds. The most influential features were  $rr(w, d)$  and Part-of-Speech (PoS) tag of the current word  $w$  in levels 1 and 2 of the Decision Tree respectively, where  $rr$  is the *repetition ratio* of the current word  $w$  in the textual ASR outputs for description  $d$ :

$$rr(w, d) = \frac{\# \text{ of ASR outputs for } d \text{ that contain } w}{\# \text{ of alternative ASR outputs for } d}.$$

### 3.2 Shallow Semantic Parser (SSP)

We found the following semantic labels useful for referring expressions:

- **Object** – a lexical item designating an object, optionally preceded by a determiner and one



or more gerunds, adjectives or nouns, e.g., “*the blue ceramic drinking mug*”.

- **Preposition** – a preposition or prepositional expression, e.g., “*on*” or “*further away from*”.
- **Landmark** – same pattern as Object, but a description may contain more than one landmark, e.g., “*the mug on the table in the corner*”.
- **Noise** – sighs or hesitations that are often misheard by the ASR as “*and*”, “*on*” or “*in*”.
- **Specifier** – a further specification that normally precedes a Landmark, e.g., “*the center of*”, “*front of*” or “*the left of*”. The preposition “*of*” at the end of a Specifier that precedes a Landmark is always required.
- **Additional** – words that are often superfluous, e.g., “*the mug that is on the table*”.

We employed the Mallet implementation of the linear chain Conditional Random Fields (CRF) algorithm (Lafferty et al., 2001) to learn sequences of semantic labels ([mallet.cs.umass.edu](http://mallet.cs.umass.edu)).

Accuracy over texts and segments was respectively measured as follows:

$$\frac{\text{\# of texts with perfectly matched label sequences}}{\text{total \# of texts}}$$
$$\frac{\text{\# of segments with perfectly matched labels}}{\text{total \# of segments}}$$

The CRF was trained separately for textual transcriptions of spoken descriptions and for ASR outputs. Two annotators labeled the 400 transcribed texts, and 800 samples from the ASR output: 400 from the best output and 400 from the worst. The first annotator segmented and labeled the descriptions, and the second annotator verified the annotations; disagreements were resolved by consensus.

We considered the features found useful in the CoNLL2001 shared task (<http://www.cnts.ua.ac.be/conll2000/chunking/>). The features that yielded the best performance were *current word*, *current PoS* and *previous word*, achieving an accuracy of 92% over the 400 textual transcriptions, and 76.13% over the 800 ASR outputs. Accuracy over segments was higher, at 96.26% for texts, and 87.28% for ASR outputs. However, SSP’s performance for the identification of Noise was rather poor, with an average accuracy of 54.75%.

### 3.3 Scusi?

*Scusi?* is a system that implements an anytime, probabilistic mechanism for the interpretation of

spoken utterances, focusing on a household context. It has four processing stages, where each stage produces multiple outputs for a given input, early processing stages may be probabilistically revisited, and only the most promising options at each stage are explored further.

The system takes as input a speech signal, and uses an ASR (Microsoft Speech SDK 6.1) to produce candidate texts. Each text is assigned a probability given the speech wave. The second stage applies Charniak’s probabilistic parser (<http://bllip.cs.brown.edu/resources.shtml#software>) to syntactically analyze the texts in order of their probability, yielding at most 50 different parse trees per text. The third stage applies mapping rules to the parse trees to generate concept graphs (Sowa, 1984) that represent the semantics of the utterance. The final stage instantiates the concept graphs within the current context. For example, given a parse tree for “*the blue mug on the table*”, the third stage returns the uninstantiated concept graph *mug(COLOR: blue) – on – table*. The final stage then returns candidate instantiated concept graphs, e.g., *mug1-location\_on-table2*, *mug2-location\_on-table1*. The probability of each instantiated concept graph depends on (1) how well the objects and relations in this graph match the corresponding objects and relations in the uninstantiated concept graph (e.g., whether *mug1* is a mug, and whether it is blue); and (2) how well the relations in this graph match the relations in the context (e.g., whether *mug1* is indeed on *table2*).

## 4 Probability Estimation

We use a distance measure inspired by the *Minimum Message Length (MML)* principle (Wallace, 2005) to estimate the goodness of a message and its semantic model. This principle is normally used for model selection, based on the following formulation:

$$\Pr(\text{data}\&\text{model}) = \Pr(\text{data}|\text{model}) \times \Pr(\text{model})$$

, which strikes a balance between model complexity and data fit, i.e., the highest-probability model that best explains the data is the best model overall. That is, the best model is not necessarily the model that fits the data best, as such a model may over-fit the data; the model itself must also have a high prior probability. In our case, the data is a text, either heard by the ASR or modified, and the model is a sequence of semantic labels. At present,

our model is restricted to semantic labels for segments in referring expressions, but in the future we will use this formalism to compare models representing different dialogue acts, e.g., commands.

Our use of the MML principle differs from its normal usage in that we employ it to compare a text and its semantic model with a modified version of this text and its own semantic model (rather than comparing two models that try to account for the same text). Modifications attract a penalty that depends on the probability that they are required (the higher the probability, the lower the penalty). This penalty is applied to prevent arbitrary modifications where a system hears what it expects.

Below we describe the estimation of the probability of a text and its semantic model. The next section describes the combination of the noisy channel model with the word-error classifier, SSP, and the modifications made to texts.

The joint probability of a Text and its Semantic Model is estimated as follows:

$$\Pr(\text{Text}\&\text{SemModel}) = \Pr(\text{Text}|\text{SemModel}) \times \Pr(\text{SemModel}),$$

where

- $\Pr(\text{SemModel}) = \Pr(\text{SSP}) \times \prod_{i=0}^{N+2} \Pr(L_i|L_0, \dots, L_{i-1}),$

where  $\Pr(\text{SSP})$  reflects SSP’s confidence in the sequence of semantic labels it produced for *Text*,  $N$  is the number of segments in the sequence,  $L_i$  is the label for segment  $i$ ,  $L_{-1}$  and  $L_0$  are the special labels **Beginning**, and  $L_{N+1}$  and  $L_{N+2}$  are the special labels **End**. To make this calculation tractable, we employ trigrams, i.e.,  $\Pr(L_i|L_0, \dots, L_{i-1}) \cong \Pr(L_i|L_{i-2}, L_{i-1}).$

- $\Pr(\text{Text}|\text{SemModel}) = \prod_{i=1}^N \Pr(\text{text}_i|L_i),$

where  $\text{text}_i$  is the sequence of words in segment  $i$ , and  $\Pr(\text{text}_i|L_i)$  is estimated as follows:

$$\Pr(\text{text}_i|L_i) = \prod_{j=1}^{M_i} \Pr(\text{HWord}_{ji}|L_i),$$

where  $M_i$  is the number of words in  $\text{text}_i$ , and  $\text{HWord}_{ji}$  is the  $j$ th heard word in  $\text{text}_i$ .

Owing to the relatively small size of our corpus,  $\Pr(\text{HWord}_{ji}|L_i)$  is roughly estimated as follows:

$$\Pr(\text{HWord}_{ji}|L_i) = \frac{\sum_{k=1}^{T_{ji}} \Pr(\text{HWord}_{ji}|\text{XpctPoS}_{kji})\Pr(\text{XpctPoS}_{kji}|L_i),$$

where  $\text{XpctPoS}_{kji}$  is a PoS expected at position  $j$  in *segment* <sub>$i$</sub> , and  $T_{ji}$  is the number of PoS expected

at position  $j$  in *segment* <sub>$i$</sub> .  $\Pr(\text{HWord}_{ji}|\text{XpctPoS}_{kji})$  is obtained from a corpus, and  $\Pr(\text{XpctPoS}_{kji}|L_i)$  is estimated from our textual transcriptions of spoken descriptions, except for the PoS associated with Noise, which are estimated from our spoken corpus (there is no Noise in texts). We obtain a rough estimate of  $\Pr(\text{XpctPoS}_{kji}|L_i)$  by considering three positions in a segment: first, middle (intermediate positions) and last. For instance, the possible PoS for the first position of an Object or Landmark are determiner, adjective, gerund, verb(past) or noun.

To illustrate this calculation, consider the second description in Table 1, which is heard as “to play it in the microwave”. The probability of the Semantic Model for this description is

$$\Pr(\text{SemModel}) = \Pr(O|B, B) \Pr(N|O, B) \Pr(P|N, O) \Pr(L|P, N) \Pr(E|L, P) \Pr(E|E, L).$$

All the probabilities involving Noise are set to an arbitrarily low  $\epsilon$ , which yields

$$\Pr(O|B, B) \Pr(E|L, P) \Pr(E|E, L) \epsilon^3.$$

The probability of the Text given the Semantic Model is

$$\Pr(\text{Text}|\text{SemModel}) = \Pr(\text{“to play”}|O) \Pr(\text{“it”}|N) \Pr(\text{“in”}|P) \Pr(\text{“the microwave”}|L),$$

which is quite high for “it”|N, “in”|P and “the microwave”|L, but is reduced due to the mismatch between the PoS of “to play” (TO VB) and the PoS expected by an Object, which are: DT/JJ/VBG/VBD/NN for the first position, and NN for the last position (Section 5.1.3).

Our system modifies this heard description by replacing “to play” with “the thing” and removing the noisy “it”, which yields “the thing in the microwave” (Section 5). The probability of the Semantic Model for this modified sentence is

$$\Pr(\text{SemModel}') = \Pr(O|B, B) \Pr(P|O, B) \Pr(L|P, O) \Pr(E|L, P) \Pr(E|E, L),$$

which is higher than that of the original Semantic Model, as is the probability of the new Text given the new Semantic Model:

$$\Pr(\text{Text}'|\text{SemModel}') = \Pr(\text{“the thing”}|O) \Pr(\text{“in”}|P) \Pr(\text{“the microwave”}|L).$$

However, this gain is offset by the penalties incurred by the modifications. The estimation of these penalties is described in the next section.

## 5 Noisy Channel Error Correction

Given a textual output produced by an ASR, we apply Algorithm 1 to remove noise, insert prepo-

sitions and replace wrong words. The probability of the resultant text and its semantic model is recalculated after each change as described in Section 4, and is moderated by the probability of the penalty for the change. Since a modification may yield a text where SSP identifies Noise, the Noise removal step is repeated after every change.

After each modification, the probability of the original text and semantic model is compared with the probability of the new text, its semantic model and any incurred penalties. The winning text and semantic model (without penalties) are then taken as the originals for the next modification. Upon completion of this process, all the incurred penalties are re-incorporated into the final probability of a modified text, in order to enable a fair comparison with other texts that were not altered.

The application of this process to all the texts produced by an ASR for a particular utterance may yield identical texts (e.g., when words with unexpected PoS are converted to “thing”). These texts are merged, and their probabilities are recalculated. The resultant texts are ranked in descending order of probability and ascending order of the number of replaced words (i.e., texts with fewer replacements are ranked ahead of texts with more replacements, irrespective of their probability). The final probabilities are adjusted to reflect the ranking of a text.

## 5.1 Estimating penalties from modifications

The modifications performed by our system attract a penalty that depends on the probability that the relevant portion of a heard utterance is wrong. The higher this probability, the lower the penalty, which is implemented as a multiplier of  $\Pr(\text{Text}\&\text{SemModel})$ .

### 5.1.1 Removing noise

The penalty for removing a heard word  $j$  in  $\text{segment}_i$  that is labeled as Noise by SSP is estimated on the basis of its probability of being Wrong (obtained from the word-error classifier, Section 3.1), as follows:

$$\Pr(\text{remove } H\text{Word}_{ji}) = \begin{cases} \Pr(\text{IsW}(H\text{Word}_{ji}))\Pr(\text{Class}) & \text{if label} = \text{W} \\ (1 - \Pr(\text{IsC}(H\text{Word}_{ji})))\Pr(\text{Class}) & \text{if label} = \text{C} \end{cases} \quad (1)$$

where  $\Pr(\text{Class})$  is the accuracy of the classifier (on training data),  $\Pr(\text{IsW}(H\text{Word}_{ji}))$  is the probability assigned by the classifier to heard word  $j$  in  $\text{segment}_i$  being Wrong, and  $\Pr(\text{IsC}(H\text{Word}_{ji}))$

---

## Algorithm 1 Noisy channel ASR error correction

---

**Require:** *Text*

```

1: SemModel ← Run SSP on Text
2: Calculate  $\Pr(\text{Text}\&\text{SemModel})$  (Section 4)
   { REMOVE NOISE }
3: while there is Noise do
4:   Text' ← Remove Noise from Text
5:   SemModel' ← Run SSP on Text'
6:   Calculate  $\Pr(\text{Text}'\&\text{SemModel}')$ 
7:    $\text{Text}\&\text{SemModel} \leftarrow \arg \max \{ \Pr(\text{Text}\&\text{SemModel}), \Pr(\text{Text}'\&\text{SemModel}')\Pr(\text{Removal}) \}$ 
8: end while
   { INSERT PREPOSITIONS }
9: while a preposition is missing do
10:  Text' ← Insert missing preposition into Text
11:  SemModel' ← Run SSP on Text'
12:  Text' ← Remove Noise from Text' (Steps 3-9)
13:  Calculate  $\Pr(\text{Text}'\&\text{SemModel}')$ 
14:   $\text{Text}\&\text{SemModel} \leftarrow \arg \max \{ \Pr(\text{Text}\&\text{SemModel}), \Pr(\text{Text}'\&\text{SemModel}')\Pr(\text{Insertion}) \}$ 
15: end while
   { REPLACE WRONG WORDS }
16: for  $i=1$  to  $N$  do
17:  Text' ← Replace wrong words in  $\text{segment}_i$ 
18:  SemModel' ← Run SSP on Text'
19:  Text' ← Remove Noise from Text' (Steps 3-9)
20:  Calculate  $\Pr(\text{Text}'\&\text{SemModel}')$ 
21:   $\text{Text}\&\text{SemModel} \leftarrow \arg \max \{ \Pr(\text{Text}\&\text{SemModel}), \Pr(\text{Text}'\&\text{SemModel}')\Pr(\text{Replacement}) \}$ 
22: end for
23:  $\Pr(\text{Text}\&\text{SemModel}) \leftarrow \Pr(\text{Text}\&\text{SemModel})\Pr(\text{Removal})\Pr(\text{Insertion})\Pr(\text{Replacement})$ 

```

---

is the probability of this word being Correct (the last two probabilities add up to 1).

The rationale for this formula is that if SSP deems a heard word to be Noise, and the classifier labels it Wrong with high probability, then its removal should cause only a small reduction in  $\Pr(\text{Text}\&\text{SemModel})$ . Conversely, if a heard word deemed to be Noise by SSP is labeled Correct by the classifier with high probability, then its removal should cause a large reduction in  $\Pr(\text{Text}\&\text{SemModel})$ . In both cases, the probabilities assigned to the labels by the classifier are moderated by the classifier’s accuracy.

To illustrate this process, let’s return to the example “to play it in the microwave”, where “it” is labeled Noise by SSP, and Wrong by the classifier with probability  $\Pr(\text{IsW}(\text{“it”}))$ . A new text *Text'* is obtained as a result of the removal of “it”, and the penalty  $\Pr(\text{IsW}(\text{“it”}))\Pr(\text{Class})$  is multiplied by the new  $\Pr(\text{Text}'\&\text{SemModel}')$ .

### 5.1.2 Inserting a preposition

If a preposition is not found in a position where one is expected, e.g., between an Object and Landmark or between an Object and a Specifier, we insert a generic preposition “at”. The penalty

for the insertion of a preposition depends on the probability that the ASR failed to hear an uttered preposition, which is estimated as follows:

$\Pr(\text{insert } P_i) = \Pr(P_i \text{ appears in } \textit{Text} \text{ and doesn't appear in the ASR output for } \textit{Text})$ , where  $P_i$  is a preposition in position  $i$  in  $\textit{Text}$ .

To determine the frequency of this event, we employ an edit distance algorithm that aligns the texts produced by the ASR with their corresponding textual transcriptions. This was done for 800 alternatives produced by the ASR (400 best and 400 worst), yielding a probability of 0.02 of the ASR dropping a preposition. The corresponding penalty for inserting a preposition (0.02) is hopefully offset by the increase in  $\Pr(\textit{SemModel}')$  as a result of this insertion. For instance, the probability of the Semantic Model for the heard description (without a preposition) in the first example in Table 1 is

$$\Pr(\textit{SemModel}) = \Pr(O|B, B) \Pr(S|O, B) \Pr(L|S, O) \Pr(E|L, S) \Pr(E|E, L),$$

where  $\Pr(S|O, B)$  and  $\Pr(L|S, O)$  are low, as they are ungrammatical. After adding the preposition,

$$\Pr(\textit{SemModel}') = \Pr(O|B, B) \Pr(P|O, B) \Pr(S|P, O) \Pr(L|S, P) \Pr(E|L, S) \Pr(E|E, L).$$

Although the new expression has an extra factor, the probabilities of the new factors are higher than those of their original counterparts.

### 5.1.3 Replacing a word

The decision to replace a word is based on the match between expected PoS and the PoS of a heard word. If they match, no replacement is performed. Otherwise, replacements are performed by applying the following rules, which are based on the PoS expected by the different types of segments at each position (first, middle, last).

- **Objects and Landmarks** – The expected PoS for Objects and Landmarks are: DT/JJ/VBG/VBD/NN for the first word, JJ/VBG/VBD/NN for the middle words, and NN for the last word. Thus, if there is a PoS mismatch, we perform the following replacements (if there is only one word in an Object or Landmark, we replace it with “thing” (NN)):

- $HWord_1 \Rightarrow$  “the” (DT)
- $HWord_{mid} \Rightarrow$  “unknown” (JJ) (multiple times)
- $HWord_{last} \Rightarrow$  “thing” (NN)

To illustrate this process, consider the heard Object “to:TO battle:NN played:VB”, which

is replaced with “the:DT battle:NN thing:NN”. Even though “battle” is incorrect, it is not modified, as its PoS is expected. However, *Scusi?* can cope with such unknown object attributes.

- **Prepositions and Prepositional Phrases** – This segment is more restricted than Objects and Landmarks, as it is largely composed of closed class words. We therefore use edit distance to find the prepositional phrase in the corpus of textual transcriptions that best matches the words in a heard prepositional phrase. The phrase from the corpus then replaces the heard segment. If there is no best-matching prepositional phrase, the generic “at” is used as a replacement. For example, “for the wave from” is replaced with “further away from” (with “from” being the next-best match), while “a all” is replaced with “at”.
- **Specifiers** – This segment is similar to Objects and Landmarks plus a final “of” when it precedes a Landmark (about 5% of the descriptions had Specifiers without Landmarks). In addition, the head noun, which is normally the penultimate word in a Specifier, must be a positional noun, such as “center”, “edge” or “corner”. Thus, a word is replaced if a PoS mismatch occurs or the penultimate word is not an expected positional noun, as follows:

- $HWord_1 \Rightarrow$  “the” (DT)
- $HWord_{mid} \Rightarrow$  “unknown” (JJ) (multiple times)
- $HWord_{last-1} \Rightarrow$  “position” (NN)
- $HWord_{last} \Rightarrow$  “of” (IN, preposition)

For instance, given the Specifier “the:DT ride:NN into:IN” followed by a Landmark, “of:IN” is appended, and “into:IN” is replaced with “position:NN”, yielding “the:DT ride:NN position:NN of:IN”. Clearly, other replacement options are possible, which will be investigated in the future.

In principle, the penalty for replacing a word should depend on both the probability that it is wrong (as for noise removal) and on the similarity between the wrong word and the proposed replacement. That is, the higher the probability that a word is wrong, and the higher the similarity between the original word and the replacement, the lower the penalty for the replacement. However, at present, we replace words that do not match an

expected PoS only with generic options, e.g., “unknown” for expected adjectives, “thing” for expected nouns in Objects and Landmarks, and “position” for expected positional nouns in Specifiers. Thus, our penalty consists only of the first of the above factors moderated by a generic similarity factor  $\delta (= 0.5)$ , as follows:

$$\Pr(\text{replace } H\text{Word}_{ji}) = \quad (2)$$

$$\begin{cases} \delta \Pr(\text{IsW}(H\text{Word}_{ji}))\Pr(\text{Class}) & \text{if label} = \text{W} \\ \delta (1 - \Pr(\text{IsC}(H\text{Word}_{ji})))\Pr(\text{Class}) & \text{if label} = \text{C} \end{cases}$$

In the future, the generic  $\delta$  will be replaced by a function of the similarity between an original word and its candidate replacements.

## 6 Evaluation

In this section, we describe our corpus and evaluation metrics, and compare the results obtained with *Scusi?* plus error correction with those obtained by the original *Scusi?* system.

### 6.1 Corpus

Our model’s performance was evaluated using part of the corpus constructed to evaluate the *Scusi?* system (Kleinbauer et al., 2013). The original corpus comprises 432 free-form descriptions spoken by 26 trial subjects to refer to 12 designated objects in four scenarios (three objects per scenario, where a scenario contains between 8 and 16 objects; participants repeated or rephrased some descriptions). 32 descriptions could not be processed by the ASR, and 105 contained constructs that could not be represented by *Scusi?*. The remaining 295 descriptions were used in our evaluation.

The descriptions, which varied in length and complexity, had an average description length of 10 words. Sample descriptions are: “the green plate next to the screwdriver at the top of the table”, “the large pink ball in the middle of the room”, “the plate on the corner of the table” and “the computer under the table”.

### 6.2 Evaluation metrics

We use the evaluation metrics discussed in (Kleinbauer et al., 2013), viz *%NotFound@N*, the percentage of descriptions that have no correct interpretation within the top  $N$  ranks; *Fractional Recall@N* (*FRecall@N*), which represents the fact that the ranked order of equiprobable interpretations is arbitrary; and *Normalized Discounted Cumulative Gain@N* (*NDCG@N*), which discounts interpretations with higher (worse) ranks (Järvelin

and Kekäläinen, 2002). The last two metrics are defined as follows:

$$F\text{Recall}@N(d) = \frac{\sum_{j=1}^N fc(I_j)}{|C(d)|},$$

where  $d$  is a description,  $C(d)$  is the set of correct interpretations for  $d$ ,  $I_j$  is an interpretation generated by *Scusi?* at rank  $j$ , and  $fc$  is the fraction of correct interpretations among those with the same probability as  $I_j$  (this is a proxy for the probability that  $I_j$  is correct):

$$fc(I_j) = \frac{c_j}{h_j - l_j + 1},$$

where  $l_j$  is the lowest rank of all the interpretations with the same probability as  $I_j$ ,  $h_j$  the highest rank, and  $c_j$  the number of correct interpretations between rank  $l_j$  and  $h_j$  inclusively.

*DCG@N* allows the definition of a relevance measure for a result, and divides this measure by a logarithmic penalty that reflects the rank of the result. Using  $fc(I_j)$  as a measure of the relevance of interpretation  $I_j$ , we obtain

$$DCG@N(d) = fc(I_1) + \sum_{j=2}^N \frac{fc(I_j)}{\log_2 j}.$$

This score is normalized to the  $[0, 1]$  range by dividing it by the score of an ideal answer where  $|C(d)|$  correct interpretations are ranked in the top  $|C(d)|$  places, yielding

$$NDCG@N(d) = \frac{DCG@N(d)}{1 + \sum_{j=2}^{\min\{|C(d)|, N\}} \frac{1}{\log_2 j}}$$

### 6.3 Results

Table 2 compares the performance of the original *Scusi?* system with that of *Scusi?* plus error correction, and displays the performance obtained for three types of modifications: N+P, P+R and N+P+R, where N stands for noise removal, P for preposition insertion, and R for word replacement (preposition insertion was folded into all the options, as it happens in only 2% of the cases). The table shows the average of *%NotFound*, *FRecall* and *NDCG* for the 295 descriptions in our corpus. The best performance is boldfaced.

As seen in Table 2, *Scusi?* plus error correction with word replacement generally outperforms the original *Scusi?* system (the Object/Landmark replacement has the greatest impact on performance among the three types of word replacements). *Scusi?*+N+P+R yields the best overall performance for *FRecall@∞* and

Table 2: Performance comparison: original *Scusi?* versus *Scusi?* + Noisy Channel Error Correction.

Average of	<i>Scusi?</i> Noisy Channel Error Correction			
		N+P	P+R	N+P+R
<i>%NotFound@∞</i>	22.37%	22.03%	14.24%	<b>13.90%</b>
<i>%NotFound@20</i>	28.14%	28.47%	<b>23.39%</b>	24.41%
<i>%NotFound@10</i>	31.86%	31.19%	<b>24.75%</b>	26.78%
<i>%NotFound@3</i>	37.97%	40.00%	<b>32.88%</b>	36.27%
<i>%NotFound@1</i>	44.75%	47.80%	<b>40.00%</b>	44.41%
<i>FRecall@∞</i>	0.776	0.778	0.858	<b>0.859</b>
<i>FRecall@20</i>	0.709	0.699	<b>0.753</b>	0.741
<i>FRecall@10</i>	0.667	0.662	<b>0.731</b>	0.712
<i>FRecall@3</i>	0.598	0.567	<b>0.636</b>	0.600
<i>FRecall@1</i>	0.488	0.462	<b>0.508</b>	0.481
<i>NDCG@∞</i>	0.641	0.626	<b>0.688</b>	0.666
<i>NDCG@20</i>	0.628	0.610	<b>0.669</b>	0.644
<i>NDCG@10</i>	0.617	0.601	<b>0.663</b>	0.636
<i>NDCG@3</i>	0.589	0.562	<b>0.624</b>	0.591
<i>NDCG@1</i>	0.516	0.490	<b>0.538</b>	0.511

*%NotFound@∞* (statistically significantly better than *Scusi?* with  $p < 0.01$  for the Wilcoxon signed rank test), while *Scusi?*+P+R yields the best performance for the remaining measures (statistically significantly better than *Scusi?* for *FRecall@∞,20,10,3*, *NDCG@∞,20,10,3* and all values of *%NotFound*; and statistically significantly better than *Scusi?*+N+P+R for *FRecall@3,1*, all values of *NDCG* and *%NotFound@3,1*,  $p \leq 0.05$ ). The fact that *Scusi?*+N+P+R outperforms *Scusi?*+P+R only for *%NotFound@∞* and *FRecall@∞* indicates that while the combination of noise removal with the other corrections enables *Scusi?* to find additional correct interpretations, these interpretations tend to appear in high (bad) ranks. The performance of *Scusi?*+N+P is generally worse than that of the original *Scusi?* system — a disappointing outcome that may be attributed to the low accuracy of SSP in the identification of Noise (54.75%, Section 3.2).

Further examination of our results reveals the following types of errors: (1) ASR errors that rendered a description unprocessable by other stages, e.g., “the green plate next to the hammer” heard as “*degree in applied next to him are*”, and “the picture above the table” heard as “the picture *of the that*”; (2) ASR errors that were not corrected, as the PoS was expected, e.g., “the center *off/IN* the table”; (3) wrong expression replacements, e.g., “the plate:O | next to *scholar of:P*” corrected as “the plate:O | next to:P”; and (4) out of vocabulary terms, e.g., “motherboard” and “frame”.

An interesting pattern emerges when considering ASR errors. Both *Scusi?*+N+P+R and *Scusi?*+P+R outperform the original version of

Table 3: Performance broken down by SER.

ASR output	Average of	<i>Scusi?</i>	P+R	N+P+R
all wrong (193 desc.)	<i>%NotFound@1</i>	61.66%	<b>52.85%</b>	54.40%
	<i>%NotFound@10</i>	44.56%	<b>33.68%</b>	35.75%
one correct (102 desc.)	<i>%NotFound@1</i>	<b>12.75%</b>	15.69%	25.50%
	<i>%NotFound@10</i>	<b>7.84%</b>	<b>7.84%</b>	9.80%

*Scusi?* for the 193 descriptions with no correct textual interpretation (SER = 65.3%, Section 1), while the original version of *Scusi?* performs at least as well as the best option, *Scusi?*+P+R, for the 102 descriptions where a correct textual interpretation was found (Table 3). This indicates that SSP is over-zealous in finding errors, and its performance must be further investigated, or another mode of operation considered (e.g., retaining both the original and the modified ASR output).

## 7 Discussion and Future Work

We have offered a noisy channel approach for error correction in spoken utterances, with a first-stage implementation that corrects errors by removing noise, inserting prepositions, and replacing wrong words with generic terms. Our approach yields significant improvements in interpretation performance, and shows promise for achieving further improvements with more sophisticated interventions.

The structure of referring expressions is rather rigid in terms of the order of the semantic segments. To test the general applicability of our noisy channel model, we propose to consider other types of dialogue acts, and take into account the expectations from the dialogue, e.g., “to play a CD” is modified when it is considered a mis-heard description, but if it were a response to the question “what would you like me to do?”, no changes would be required. In addition, we will extend our approach to propose specific, rather than generic, word replacements, and to handle superfluous information (i.e., information that is meaningless to the language interpretation module) or missing information (e.g., missing landmarks). Another avenue of research involves versions of *Scusi?* that employ SSP as an alternative to or in combination with a syntactic parser.

## Acknowledgments

This research was supported in part by grants DP110100500 and DP120100103 from the Australian Research Council. The authors thank Masud Moshtaghi for his help with statistical issues.

## References

- C.F. Baker, C.J. Fillmore, and J.B. Lowe. 1998. The Berkeley FrameNet project. In *COLING'98 – Proceedings of the 17th International Conference on Computational Linguistics*, pages 86–90, Montreal, Canada.
- E. Brill and R.C. Moore. 2000. An improved error model for noisy channel spelling correction. In *ACL'2000 – Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong.
- B. Coppola, A. Moschitti, and G. Riccardi. 2009. Shallow semantic parsing for spoken language understanding. In *NAACL-HLT 2009 – Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 85–88, Boulder, Colorado.
- P. Domingos and M. Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130.
- J. Geertzen. 2009. Semantic interpretation of dutch spoken dialogue. In *IWCS-8 – Proceedings of the 8th International Conference on Computational Semantics*, pages 286–290, Tilburg, The Netherlands.
- K. Järvelin and J. Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.
- M. Johnson and E. Charniak. 2004. A TAG-based noisy channel model of speech repairs. In *ACL'04 – Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 33–39, Barcelona, Spain.
- Th. Kleinbauer, I. Zukerman, and S.N. Kim. 2013. Evaluation of the *Scusi?* spoken language interpretation system – A case study. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, Nagoya, Japan.
- J.D. Lafferty, A. McCallum, and F.C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'2001 – Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, Massachusetts.
- R. López-Cózar and D. Griol. 2010. New technique to enhance the performance of spoken dialogue systems based on dialogue states-dependent language models and grammatical rules. In *Proceedings of Interspeech 2010*, pages 2998–3001, Makuhari, Japan.
- T. Pellegrini and I. Trancoso. 2010. Improving ASR error detection with non-decoder based features. In *Proceedings of Interspeech 2010*, pages 1950–1953, Makuhari, Japan.
- J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California.
- E. Ringger and J.F. Allen. 1996. Error correction via a postprocessor for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 427–430, Atlanta, Georgia.
- T.N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky. 2011. Exemplar-based sparse representation features: From TIMIT to LVCSR. *IEEE Transactions on Audio, Speech and Language Processing*, 19(8):2598–2613.
- J.F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.
- C.S. Wallace. 2005. *Statistical and Inductive Inference by Minimum Message Length*. Springer, Berlin, Germany.
- I. Zukerman, E. Makalic, M. Niemann, and S. George. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 581–592, Hanoi, Vietnam.
- I. Zukerman, P. Ye, K.K. Gupta, and E. Makalic. 2009. Towards the interpretation of utterance sequences in a dialogue system. In *Proceedings of the 10th SIGDial Conference on Discourse and Dialogue*, pages 46–53, London, United Kingdom.
- S. Zwartz, M. Johnson, and R. Dale. 2010. Detecting speech repairs incrementally using a noisy channel approach. In *COLING'2010 – Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1371–1378, Beijing, China.

# Natural Language Query Refinement for Problem Resolution from Crowd-Sourced Semi-Structured Data

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy

IBM Research,

India Research Lab

{rashgang,murali.balakrishnan}@in.ibm.com

## Abstract

We study the problem of natural language query generation for decision support systems (DSS) in the problem resolution domain. In this domain, a user has a task he is unable to accomplish (eg. *bluetooth headphone not playing music*), which we capture using language structures. We show how important units that define a problem can robustly and automatically be extracted from large noisy online forum data, with no labeled data or query logs. We also show how these units can be selected to reduce the number of interactions and how they can be used to generate natural language interactions for query refinement.

## 1 Introduction

Decision Support Systems (DSSs) that help decision makers extract useful knowledge from large amounts of data have found widespread application in areas ranging from clinical and medical diagnosis (Musen et al., 2006) to banking and credit verification (Palma-dos Reis et al., 1999). IBM's Watson Deep Question Answering system (Ferrucci et al., 2010), can be applied to DSSs to diagnose and recommend treatments for lung cancer and to help manage health insurance decisions and claims<sup>1</sup>. Motivated by the rapid explosion of contact centres, we focus on the application of DSSs to assist technical contact center agents.

Contact center DSSs should be designed to assist an agent in the *problem resolution* domain. This domain is characterized by a user calling in to a contact center with the problem of being unable to perform an action with their product (e.g. *I am unable to connect to youtube*). Currently contact center DSSs are essentially search engines for

technical manuals. However, this has two shortcomings : (i) in most cutting edge consumer technology, like software and smart devices, the range of possible applications and use cases makes it impossible to list all of them in the manuals- limiting their usefulness under the heavy tailed nature of customer problems, (ii) contact centers are known to suffer from high churn due to pressures and difficulties of the jobs, particularly the need for rapid resolution, making ease of use essential since users of these DSSs are somewhere between experts (in using the system) and novices (in the actual technology customers need help with).

With the birth and growth of the Web 2.0 and in particular, large and active online product forums, such as, Yahoo! Answers<sup>2</sup>, Ubuntu Forums<sup>3</sup> and Apple Support Communities<sup>4</sup>, there is the hope that other technology savvy users will find and resolve large number of problems within days of the release of a product. However, these forums are *noisy*, i.e. they contain many throw-away comments and erroneous solutions. The first important question we address in this paper is, how can we mine relevant information from online forums and, in essence, *crowdsource* the creation of contact center DSSs? In particular, we show how many problems faced by consumers can be captured by *actions on attributes* (e.g. *bluetooth headphone not playing music*).

In order to address the second shortcoming, we study the problem of automatic *interactive* query refinement in DSSs. When DSSs are used by non-computer scientists, natural language understanding and interaction problems take center-stage (Alter, 1977). Since both customers and agents are not experts in a technical area, mis-understandings are common. As agents are evaluated based on the number of problems resolved, it is often the case

<sup>1</sup><http://www.ihealthbeat.org/articles/2013/2/11/ibm-offering-two-new-decision-support-tools-based-on-watson.aspx>

<sup>2</sup><http://answers.yahoo.com/>

<sup>3</sup><http://ubuntuforums.org/>

<sup>4</sup><https://discussions.apple.com/>



that queries entered by an agent are underspecified. In response to such a query, a search engine may return a large number of documents. For complicated technical queries, the time taken by an agent to read the long list of returned information and possibly reformulate the query could be significant. The second question we address in this paper is how can a DSS make *natural language* suggestions that assist the agent in acquiring additional information from a caller to resolve her problem in the shortest amount of time? Finally, for rapid prototyping and deployment, we develop a system and architecture that *does not use any form of labeled data or query logs*, a big differentiator from prior work. Query Logs are not always available for enterprise systems that are not on the web and/or have a smaller user base (Bhatia et al., 2011). When software and hardware change rapidly over time, it is infeasible to quickly collect large query logs. Also, logs may not always be accessible due to privacy and legal constraints.

To the best of our knowledge, this paper presents the first interactive system (and detailed evaluation thereof) for natural language problem resolution in the absence of manually labeled logs or pre-determined dialog state sequences. Concretely, our primary contributions are:

- **Problem Representation and Unit Extraction:** We define and automatically extract units that best represent a problem. We show how to do this robustly from noisy forum threads, allowing us to use abundant online user generated content.
- **Unit selection for Interaction:** We propose and evaluate a complete interactive system for users to quickly find a resolution based on semi-structured online forum data. Follow up questions are generated automatically from the retrieved results to minimize the number of interactions.
- **Natural Language Question Generation:** We demonstrate that, in a dialog system it is possible and useful to automatically generate fluent interactions based on the units we define *using appropriate templates*. We use these to create follow up questions to the user, which have much needed context, and show that this improves precision.

## 2 Proposed System

In online forums, people facing issues with their product (thread initiators) post their problems and other users write subsequent posts discussing solutions. These threads form a rich data source that

could contain problems similar to what a user who calls in to the contact center faces, and can be used to find an appropriate solution. Our system (Figure 1) has two phases. In the offline phase, the system extracts units that describe the problem being discussed. In the online phase, the interaction engine selects the most appropriate units that best divide the space of search results obtained, to minimize the number of interactions. The system then generates follow up interactions by inserting the units into appropriate unit type dependent templates. The answers to these follow up questions are used to improve the search results.

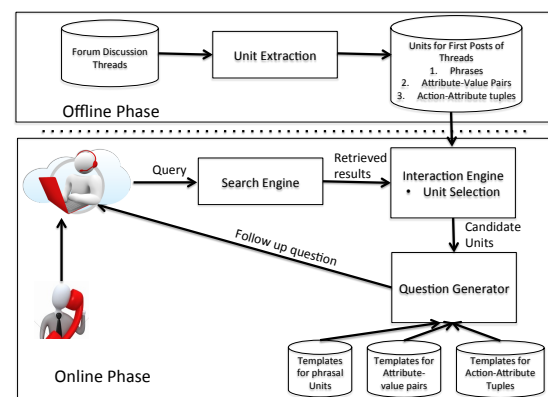


Figure 1: System Description

### 2.1 Representational Units

It is important to select representational units that capture the signature, or the most important characteristics of the information that users search for. This signature should be sufficient to find relevant results. In order to understand the right units for the problem resolution domain, we conducted the following user study. Five annotators analyzed the first posts from 50 threads from the Apple Discussion Forum, and were asked to mark the most relevant short segments of the post that best described the problem (an example in Table 1).

<p>I cannot hear the notifications on my bluetooth now. it's at normal volume when i send a message, if i receive an email or text volume is very low yet I have all volumes up all the way. Is there a new bluetooth volume i have to turn up? or did another update screw the bluetooth . Was working just great before i updated to ios - 4 . please help me.</p>
--

Table 1: Relevant short segments for a forum post.

Based on the user study, the first kind of units we considered were **phrases**, which are consecutive words that occur very frequently in the thread.

Phrases as query suggestions have been shown to improve user experience when compared to just showing terms (Kelly et al., 2009) since longer phrases tend to provide more context information. One shortcoming of these contiguous phrasal units is that they are sensitive to typography, i.e. small changes in phrasing (e.g. *ios - 4* and *ios 4*) lead to different phrases and the occurrence counts are divided among these variations. This causes difficulties both in the problem representation as well as in the search for problem resolution which are exacerbated by the noisy, casual syntax in forums. Motivated by Probst et al. (2007), we extract **attribute-value pairs**. These units provide both robustness as well as more configurational context to the problem. Another observation from the segments marked was that many of them involved a user wanting to perform an action (*I cannot hear notifications on bluetooth*) or the problems caused by a user’s action (*working great before I updated*). We capture them using **action-attribute tuples** (details in Section 3.1.1).

Thread initiators describe the same problem in different ways leading to multiple threads discussing the same problem. Ideally, we want the representation of the problem to be the same for all these threads to build robust statistics. Consider the following examples, *sync server has failed*, *sync server failed*, *sync server had failed*, *sync server has been failing*. While the phrasing is different, we see that their dependency parse trees (Figure 2) show a common relation between the verb or action, *fail*, and the attribute *sync server* (the base form of the verbs are obtained from their corresponding lemmas with TreeTagger (Schmid, 1994)). Motivated by this, we use dependency parse trees for extracting action-attribute tuples.

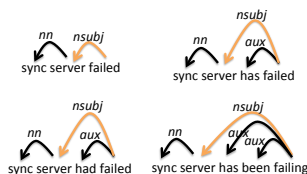


Figure 2: Dependency Parse Trees for various forms of the same problem.

### 3 Detailed System Description

We now give a detailed description, showing how the three types of units can be robustly extracted and used from noisy online forums.

## 3.1 Offline Component

In the offline phase we first extract candidate units that describe a problem (and its solution) from the forum threads. We then filter this description, using the thread itself, to retain the important units.

### 3.1.1 Candidates for Problem Description

Sentences are tagged and parsed using the Stanford dependency parser (de Marneffe et al., 2006). The following units are then obtained from the first post of the discussion thread.

**Phrasal units:** defined to be Noun Phrases satisfying the regular expression,  $(Adjective)^* (Number/Noun/Proper\ Noun)^+$ , (eg., *interactive web sites*, *osx widgets*, *2007 outlook calendar*). These are extracted from the discussion thread along with their frequencies of occurrence.

**Attribute-Value pairs:** The dependency relations *amod* (adjectival modifier of a noun phrase) and *num* (numeric modifier of a noun) in the parsed output are used for this purpose. In the case of *amod*, the attribute is the noun that the adjective modifies and its value is the adjective. For example, with *amod(signal, strong)*, the attribute is *signal* and its value is *strong*. In the case of *num*, the attribute is the noun and its value is the numeric modifier. For example, with *num(digits, 10)*, the attribute is *digits* and its value is *10*. As mentioned in Section 2.1, these pairs capture more context of the problem being discussed. Additional attribute-value pairs are extracted by expanding the attributes with the adjacent nouns and adjectives that occur with it. For the example in Figure 3, the attribute *signal* is modified to *cell phone signal* and added to the list of attribute-value pairs along with their frequencies of occurrence.

**Action-Attribute tuples:** The dependency relations used for these units are given in Table 3 with examples. Many of these units help describe the user’s problem while others provide contextual information behind the problem being discussed. These units are described with 4-tuples  $(Arg_1-verb-Arg_2-Arg_3)$ , three of which are the arguments of the verb or attributes of an action. The relations given in the first column of Table 3 form fillers for the attributes of the action. The example in Figure 3 gives the tuple, *I-entered-dns-null*, where, *I* is the subject, *entered* is the action performed, *dns* is the object. If the verb has a *prt* relation, the particle is appended to the verb. For example, *turned* has a *prt* relation in *prt(turned, off)*,

First Post	I cannot hear the notifications on my bluetooth now. it's at normal volume when i send a message but if i receive an email or text the volume is very low yet I have all volumes up all the way. Is there a new bluetooth volume i have to turn up with ios - 4? or is it that another update screwed with the bluetooth again. Was working just great before i updated ios - 4. please help me. thanks!
Phrases	volume, bluetooth volume, bluetooth, notifications, update, ios, normal volume, work, text, way, email, message, new bluetooth volume,
Phrases + Att-Val pairs	volume, bluetooth volume, bluetooth, notifications, update, ios, normal volume, work, text, way, email, message, new bluetooth volume, ios 4
Phrases + Att-Val pairs + Act-Att tuples	volume, bluetooth volume, bluetooth, notifications, update, ios, normal volume, work, text, way, email, message, new bluetooth volume, ios 4, low volume, I hear notifications on bluetooth, update screwed bluetooth, I send message, I receive email, it is at normal volume, working great before updated, I missed emails, *I volumes way.

Table 2: Problem representations for a forum post.

hence, the verb is now modified to *turned off*. Since *entered* in this example takes only a subject and an object as arguments, the third argument is *null*. Consider another example, *I removed the wep password in the router settings*, the tuple is now *I-removed-password-in the settings*. The last row in Table 3 gives an example of the usage of the *xcomp* relation. As done with Attribute-Value pairs, the attributes in these units are also expanded with the adjacent nouns and adjectives and added to the list of Action-Attribute tuples along with their frequencies of occurrence in the entire thread.

### 3.1.2 Scoring and Filtering

Since the problem is defined by the thread initiator in the first post of the thread, units in the first post are scored and ranked based on tf-idf (Manning et al., 2008). We treat each thread as a document and the top 50 highest scoring candidates form the problem description for the thread. Units are extracted from the rest of the thread in order to obtain frequency statistics for the units in the first post. Pronouns, prepositions and determiners are dropped from the units while obtaining the counts. In addition, verbs in the action-attribute tuples are converted to their base form using the lemma information obtained from TreeTagger (Schmid, 1994), to obtain counts. This makes the scores robust to small variations in the units.

Examples of extracted units are shown in Table 2. We see that errors in the parse (*volumes* was tagged as a verb) cause erroneous units (*\*I volumes up*). For this reason, we use frequency statistics from the rest of the discussion thread, to determine if a unit is valid or not. The tf-idf based scheme also removes commonly used phrases such as, *please help me, thank you*, etc.

## 3.2 Online Phase

The system searches for a set of initial documents based on the user's initial query. Next, the follow-up candidate units are selected (Section 3.2.1) from the units extracted in the offline phase for the

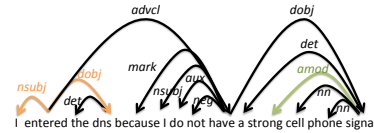


Figure 3: Dependency parse: *I entered the dns because I do not have a strong cell phone signal*.

retrieved documents and natural language interactions are further generated by filling the templates (Section 3.2.2) with the selected units.

### 3.2.1 Selection of candidate units for Question Generation

Interactions should be selected to (i) understand the user's requirements better by making the query more specific and reduce the number of results returned by the search engine, and (ii) reduce the number of interactions required. We use information gain to find the best unit that reduces the search space, motivated by its near optimality (Golovin et al., 2010). If  $S$  is the set of all retrieved documents,  $S_1 \subseteq S$  containing  $unit_i$  and  $S_2$  is a subset of  $S$  that do not contain  $unit_i$ , the gain from branching (or interacting) on  $unit_i$  is,

$$Gain(S, unit_i) = E(S) - \frac{|S_1|}{|S|} E(S_1) - \frac{|S_2|}{|S|} E(S_2) \quad (1)$$

$$E(S) = \sum_{k=1, \dots, |S|} -p(doc_k) \log_2 p(doc_k) \quad (2)$$

Where, each document is assigned a probability based on its rank in the search results:

$$p(doc_j) = \frac{\frac{1}{rank(doc_j)}}{\sum_{k=1, \dots, |S|} \frac{1}{rank(doc_k)}} \quad (3)$$

The unit that gives the highest information gain forms the candidate for question generation. Information gain has been widely used in Decision Trees (Mitchell, 1997), where the nodes represent attributes and the edges indicate values, and is known to result in short trees. In our case, the nodes represent the follow up questions and the edges indicate whether the user's answer is *yes* or

Relation	Example	Attributes(Arg <sub>1</sub> /Arg <sub>2</sub> /Arg <sub>3</sub> )	Action(Verb)
nsubj	nsubj(entered,I)	I (Arg <sub>1</sub> )	entered
dobj	dobj(entered,dns)	dns (Arg <sub>2</sub> )	entered
iobj	iobj(give, address)	address (Arg <sub>3</sub> )	address
pobj	prep(connect,to), pobj(to,wifi)	wifi(Arg <sub>2</sub> )	connect
prep_(to,into, etc)	prep_in(removed,settings)	in the settings (Arg <sub>3</sub> if Arg <sub>2</sub> not present, else Arg <sub>2</sub> )	removed
xcomp	xcomp(prompt, connect), prep_to(connect,wifi), nsubj (prompt, password)	password(Arg <sub>1</sub> ), to wifi (Arg <sub>2</sub> )	prompt to connect

Table 3: Dependency relations used to extract Action-Attribute tuples

*no*. The goal in decision trees is to quickly exhaust the space of examples with fewer steps, resulting in shorter trees. The goal in this paper is to traverse the space of results obtained with the initial query to reach the most relevant document with the fewest interactions. Since the dialog problem can be easily mapped to decision trees, the choice of information gain allows the user to arrive at the most relevant document with the smallest number of interactions in the online phase.

### 3.2.2 Question Generation

Questions are generated based on the type, number and tense information present in the units. The list of templates used for question generation is given in Table 4. Once a candidate unit is selected, a template is chosen based on its type. Phrasal units have a single template. If an attribute has two values with very similar information gains, the template for Attribute-Value pairs accommodates the different values. For example, if the pairs, *Outlook:2003* and *Outlook:2007* have very similar gains, the question would then be *Is your outlook: Option<sub>1</sub>:2003 Option<sub>2</sub>:2007 ?* and the user has the option to click on the one that is relevant to his query. For Action-Attribute tuples, the templates are chosen based on the the person, number and tense information from the verbs (Table 4). *null* in the table (for example, *null-send-emails-null*) indicates that a particular argument does not exist or was not found and hence the argument will not be added to the appropriate template. Certain templates require converting the verb to a different form (e.g., *VBD* to *VBN*). This mapping is stored as a dictionary obtained by running the TreeTagger on the entire dataset and various forms are automatically obtained by linking them to the lemmas of the verbs (for example, *give(VB/lemma) gave(VBD) given(VBN) gives(VBZ)*).

## 4 Results and Discussion

To evaluate our system, we built and simulated a contact center DSS for iPhone problem resolution.

### 4.1 Description of Dataset

We crawled threads created during the period 2007-2011 from the Apple Discussion Forum resulting in about 147,000 discussion threads. In order to create a test data set, threads were clustered treating each discussion thread as a data point using a tf-idf representation. The thread nearest the centroid from the 60 largest clusters were marked as the ‘most common’ problems.

<p>Underspecified Query 1: “cannot sync calendar”  Forms 6 specific queries</p> <ol style="list-style-type: none"> <li>1. because iphone disconnected</li> <li>2. because the sync server failed to sync</li> <li>3. because the sync session could not be started</li> <li>4. because the phone freezes</li> <li>5. error occurred while mingling data</li> <li>6. error occurred while merging data</li> </ol>
--

Table 5: Specific and under-specified queries

To generate specific and under-specified queries on this data set, in our experiments, we use the first post as a proxy for the problem description. An annotator created one short query (underspecified) from the first post of each of the 60 selected threads. These queries were given to the Lemur search engine (Strohman et al., 2005) to retrieve the 50 most similar threads from an index built on the entire set of 147,000 threads. He manually analyzed the first posts of the retrieved threads to create contexts, resulting in 217 specific queries. To understand this process we give an example from our data creation in Table 5. Starting from an under-specified query *cannot sync calendar*, the annotator found 6 specific queries. Two other annotators, were given these specific queries along with the search engine’s results from the corresponding under-specified query. They were asked to choose the most relevant results for the specific queries. The intersection of the choices of the annotators formed our ‘gold standard’.

### 4.2 User based analysis of the problem representation and unit extraction

We conducted two user studies to determine the (subjective) value of our problem representation,

Unit's Type	Template (with examples)
Phrases	Is your query related to <i>[unit]</i> ? eg: Is your query related to <i>osx widgets</i> ?
Attribute-Value pairs (if single value)	Is your <i>[attribute]</i> <i>[value]</i> ? eg: Is your <i>wifi signal strong</i> ?
Attribute-Value pairs (if multiple values)	Is your <i>[attribute]</i> : Option <sub>1</sub> : <i>[value<sub>1</sub>]</i> ... Option <sub>n</sub> : <i>[value<sub>n</sub>]</i> ? eg: Is your <i>outlook calendar</i> : Option <sub>1</sub> : <i>2003</i> Option <sub>2</sub> : <i>2007</i> ?
Action-Attribute tuples (Verb/Action is VB: base form)  ARG <sub>1</sub> is empty / ARG <sub>1</sub> is a pronoun	Does the <i>[ARG<sub>1</sub>(sg)] [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Do the <i>[ARG<sub>1</sub>(pl)] [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Do you want to <i>[VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? eg: Does the <i>wifi network prompt</i> the <i>password</i> ? from the tuple: <i>wifi network-prompt-password-null</i>
Action-Attribute (Verb/Action is VBP: non-3rd person, singular, present) ARG <sub>1</sub> is empty / ARG <sub>1</sub> is a pronoun	<i>[ARG<sub>1</sub>] [VERB] [ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Do you want to <i>[VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? eg: Do you want to <i>send</i> the <i>emails</i> ? from the tuple: <i>null-send-emails-null</i>
Action-Attribute (Verb/Action is VBN: past participle) ARG <sub>1</sub> is empty / ARG <sub>1</sub> is a pronoun ARG <sub>2</sub> and ARG <sub>3</sub> are empty ARG <sub>2</sub> and ARG <sub>3</sub> are empty	Have you <i>[VERB] [ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Has the <i>[ARG<sub>1</sub>(sg)]</i> been <i>[VERB]</i> ? Have the <i>[ARG<sub>1</sub>(pl)]</i> been <i>[VERB]</i> ? Has the <i>[ARG<sub>1</sub>(sg)] [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Have the <i>[ARG<sub>1</sub>(pl)] [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? eg: Has the <i>update caused</i> the <i>phone to crash</i> ? from the tuple: <i>update-caused-phone-to crash</i>
Action-Attribute (Verb/Action is VBZ: 3rd person, singular, present) ARG <sub>1</sub> is empty	Does the <i>[ARG<sub>1</sub>] [VERB<sub>VB</sub>]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Does the <i>phone [VERB<sub>VB</sub>]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? eg: Does the <i>iphone use idol support</i> from the tuple: <i>iphone-uses-idol support-null</i>
Action-Attribute (Verb/Action is VBD: past tense) ARG <sub>1</sub> is empty ARG <sub>1</sub> is empty ARG <sub>1</sub> is a pronoun	Has the <i>[ARG<sub>1</sub>(sg)] [VERB<sub>VBN</sub>] [ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Have the <i>[ARG<sub>2</sub>(pl)] [ARG<sub>3</sub>]</i> been <i>[VERB<sub>VBN</sub>]</i> ? Is the <i>[ARG<sub>2</sub>(sg)] [ARG<sub>3</sub>] [VERB<sub>VBN</sub>]</i> ? Have you <i>[VERB<sub>VBN</sub>] [ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Have the <i>[ARG<sub>1</sub>(pl)] [VERB<sub>VBN</sub>] [ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? eg: Has the <i>iphone found several networks</i> ? from the tuple: <i>iphone-found-several networks-null</i>
Action-Attribute (Verb/Action is VBG: gerund/present participle)  ARG <sub>1</sub> is empty	Is the <i>[ARG<sub>1</sub>(sg)] [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Are the <i>[ARG<sub>1</sub>(pl)] [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? Is the <i>phone [VERB]</i> the <i>[ARG<sub>2</sub>] [ARG<sub>3</sub>]</i> ? eg: Is the <i>site delivering</i> the <i>flash version</i> ? from the tuple: <i>site-delivering-flash version-null</i>

Table 4: Templates for the follow up Question Generation .

focusing on action-attribute tuples. In the first user study, 5 users were given the first post of 20 threads with three problem representations for the first post, (1) phrasal units only, (2) phrasal units and attribute-value (Att-Val) pairs and (3) phrasal units, attribute-value pairs and action-attribute (Act-Att) tuples. They were asked to indicate which representation best represented the problem. All users preferred the third representation on all the first posts. An example first post and units are in Table 2.

In the second study, the same 5 users were asked to indicate how many units in Representation 3 were not relevant to the problem discussed in the first post, for a subset of 10 threads. We defined ‘not relevant’ as noisy components which do not aid in the problem representation e.g. *oh boy* and *thanks!* (see Table 2). All users marked 2 examples

(*sort* and *way*) as not relevant, out of 110 units that the algorithm generated for these threads.

These two user studies taken together show that the combined set of units, is able to capture the problem description well and that our algorithm is able to filter out noise in the thread to create a robust and useful representation of the problem. The results in Section 4.3 (Tables 6 and 7), show the value of our problem representation, in a complete end-to-end system, with objective metrics.

### 4.3 Unit selection for interaction

We evaluate a complete system with both user (or agent) and search engine in the loop. We focus on measuring the value of the interactions by an analysis of which results ‘rise to the top’. The experiment was conducted as follows. Annotators were given a specific query and its underspecified query



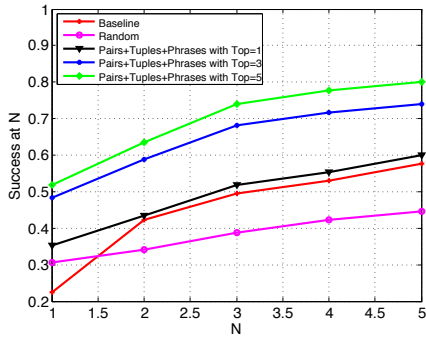


Figure 4: Success at N.

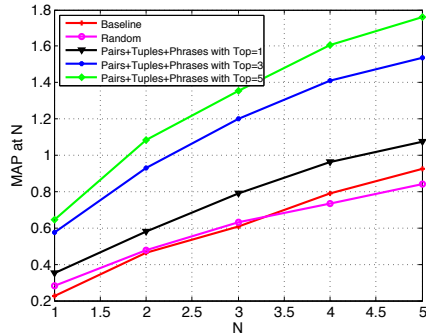


Figure 5: MAP at N.

(as created in Section 4.1) along with the results obtained when the underspecified query was input to the search engine. They were presented with the  $Top = 1, 3$  or  $5$  scoring follow up questions. E.g., for the underspecified query in Table 5 and specific query 2, the generated question was (see Table 4), *Has the sync server failed to sync?*. The user then selected the most appropriate follow up question, reducing the number of results. We then measured the relevance of the reduced result, with respect to the gold standard (see Section 4.1) for that specific query, using metrics commonly used in Information Retrieval - MRR, Mean Average Precision (MAP) and Success at rank  $N$  (Baeza-Yates et al., 1999). We restrict  $N = 5$  (small) since the rapid resolution time required of contact center agents does not allow them to look at many results.

In Figures 4, 5 and Table 6, we compare our system against a baseline system, which is the set of results obtained with the underspecified query, and a system where 5 interaction units are selected at random from the initial search results. Note that, as the number of follow up questions presented increases, the scores will improve since it is more likely that the ‘correct’ choice is presented. However, there is a trade-off here since the agent has to again peruse more questions, which increases

time spent, and so we limit this value to 5 as well. In terms of all three measures, our system is able

Unit’s Type	MRR
Baseline	0.3997
Random	0.4021
Phrases (with Top=5)	0.6548
Phrases, Pairs (with Top=5)	0.6745
Phrases, Pairs, Tuples (with Top=5)	0.7362

Table 6: MRR for different unit types.

to give a substantial improvement in performance. E.g., one intelligently chosen interaction performs better than 5 randomly chosen ones. These results show the value of the units we select and the choice of information gain as a metric. To measure the importance of each unit type, we analyzed the selected follow up questions ( $Top = 5$ ) for each underspecified query. Table 7 lists the fraction of queries whose origin was a specific unit type.

Unit’s Type	Preference
Phrases	51%
Attribute-Value Pairs	12%
Action-Attribute Tuples	37%

Table 7: Fraction of follow up questions selected that originated from a specific unit type

#### 4.4 Evaluating Templates for Question Generation

Finally, an annotator was given 100 generated follow up questions from the previous experiment and asked to label them as understandable or not. The annotator marked 13% as not understandable. Examples were, *does the phone connect*, *has the touchscreen stopped*, *does the message connect* (which were due to errors in parsing) and *do you want to leave the car* (due to a filtering error).

## 5 Related Work

Our work is related to three somewhat distinct areas of research, dialog systems, question answering (QA) systems and interactive search. Unlike most QA systems, we continue a sequence of interactions where the system and the user are active participants. The primary contribution of this work is a combined DSS, search, natural language dialog and query refinement system built automatically from semi-structured forum data. No prior work on interactive systems deals with problem resolution from large scale, noisy online forums.

Many speech dialog systems exist today for tasks including, obtaining train information (RAILTEL) (Bennacef et al., 1996), airline information (Rudnicky et al., 2000) and weather information (Zue et al., 2000). These systems perform simple database retrieval tasks, where, the keywords and their possible values are known apriori.

In general document retrieval tasks, when a user's query is under-specified and a large number of documents are retrieved, interactive search engines have been designed to assist the user in narrowing down the search results (Bruza et al., 2000). Much research has concentrated on query reformulation or query suggestions tasks. Suggestions are often limited to terms or phrases either extracted from query logs (Guo et al., 2008; Baeza-Yates et al., 2004) or from the documents obtained in the initial search results (Kelly et al., 2009). Bhogal et al. (2007) require rich ontologies for query expansion, which may be difficult and expensive to obtain for new domains. Leung et al. (2008) identify related queries from the web snippets of search results. Cucerzan and White (2007) use users' post-query navigation patterns along with the query logs to provide query suggestions. Mei et al. (2008) rank query suggestions using the click-through (query-url) graph. Boldi et al. (2009) provide query suggestions based on short random walk on the query flow graph. The main drawback behind these approaches is the dependence on query logs and labeled data to train query selection classifiers. We show how certain units are robust representations of documents in the problem resolution domain which can automatically be extracted from semi-structured data.

Feuer et al. (2007) use a proximity search-based system that suggests sub and super phrases. Cutting et al. (1993; Hearst and Pedersen (1996; Kelly et al. (2009) cluster retrieved documents and make suggestions based on the centroids of the clusters. Kraft and Zien (2004) and Bhatia et al. (2011) use  $n$ -grams extracted from the text corpus to suggest query refinement. Although these techniques do not rely on query logs for providing suggestions, the suggestions are limited to *contiguous* phrases. They also do not generate follow up questions, but instead provide a list of suggestions and require the user to select one among them or use them manually to reformulate the initial queries.

Automatically framing natural language questions as follow up questions to the user is still a

challenging task since, (1) Diriye et al. (2009) and Kelly et al. (2009) showed that interactive query expansion terms are poorly used, and tend to lack information meaningful to the user, thus emphasizing the need for larger context to best capture the actual query/problem intent (2) finding a few question/suggestions that would narrow the search results will lead to fewer interactions as opposed to displaying the single best result (3) particularly for non-technical users, interactions and clarifications need to be fluent enough for the user to understand and continue his interaction with the system (Alter, 1977). In this paper, we show how to extract important representative contextual units (which do not necessarily contain contiguous words) and use these to generate contextual interactions.

Sajjad et al. (2012) consider a data set where objects belong to a known category, with textual descriptions of objects and categories collected from human labelers, using which  $n$ -gram based attributes of objects are defined. Subsets of these attributes are filtered, again using labeled data. Kotov and Zhai (2010) frame questions with the help of handmade templates for the problem of factoid search from a subset of Wikipedia. However, they do not select queries with the goal of minimizing the number of interactions. To extend these approaches to problem-resolution finding, (as opposed to factoids or item descriptions) simple most common noun phrases (as used in Sajjad et al. (2012) and Kotov and Zhai (2010)) are insufficient, since they do not capture the problem or intent of the user. As motivated in Section 1, this requires a better representation of candidate phrases. Our paper also suggests an approach that does not need any human labelled or annotated data. Suggestions are selected using units such that the problem intent is well captured and also ensure that fewer interactions take place between the user and the system. Follow-up questions are framed using templates designed for these units, allowing us to move beyond simple terms and phrases.

## 6 Conclusion and Future Work

This paper proposed an interactive system for natural language problem resolution in the absence of manually labelled logs or pre-determined dialog state sequences. As future work, we would like to use additional information such as, the trustworthiness of the posters, quality of solutions in the threads, etc., while scoring the documents.

## References

- Steven Alter. 1977. Why is man-computer interaction important for decision support systems? *Interfaces*, 7(2):109–115.
- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. ACM press.
- Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2004. Query recommendation using query logs in search engines. In *EDBT*.
- S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. 1996. Dialog in the RAILTEL telephone-based system. In *ICSLP*.
- Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query suggestions in the absence of query logs. In *SIGIR*.
- J. Bhogal, A. Macfarlane, and P. Smith. 2007. A review of ontology based query expansion. *Inf. Process. Manage.*
- Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, and Sebastiano Vigna. 2009. Query suggestions using query-flow graphs. *WSCD*.
- Peter Bruza, Robert McArthur, and Simon Dennis. 2000. Interactive internet search: keyword, directory and query reformulation mechanisms compared. In *SIGIR*.
- Silviu Cucerzan and Ryen W. White. 2007. Query suggestion based on user landing pages. In *SIGIR*.
- Douglass R. Cutting, David R. Karger, and Jan O. Pedersen. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In *SIGIR*.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. *LREC*.
- Abdigani Diriye, Ann Blandford, and Anastasios Tombros. 2009. A polyrepresentational approach to interactive query expansion. In *ACM/IEEE-CS*.
- 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).
- Alan Feuer, Stefan Savev, and Javed A Aslam. 2007. Evaluation of phrasal query suggestions. In *CIKM*.
- Daniel Golovin, Andreas Krause, and Debajyoti Ray. 2010. Near-optimal bayesian active learning with noisy observations. In *NIPS*.
- Jiafeng Guo, Gu Xu, Hang Li, and Xueqi Cheng. 2008. A unified and discriminative model for query refinement. In *SIGIR*.
- Marti A. Hearst and Jan O. Pedersen. 1996. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR*.
- Diane Kelly, Karl Gyllstrom, and Earl W. Bailey. 2009. A comparison of query and term suggestion features for interactive searching. In *SIGIR*.
- Alexander Kotov and ChengXiang Zhai. 2010. Towards natural question guided search. In *WWW*.
- Reiner Kraft and Jason Zien. 2004. Mining anchor text for query refinement. In *WWW*.
- Kenneth Wai-Ting Leung, Wilfred Ng, and Dik Lun Lee. 2008. Personalized concept-based clustering of search engine queries. *IEEE Trans. on Knowl. and Data Eng.*
- Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.
- Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. 2008. Query suggestion using hitting time. In *CIKM*.
- Tom M Mitchell. 1997. *Machine learning*. Burr Ridge, IL: McGraw Hill.
- Mark A Musen, Yuval Shahar, and Edward H Shortliffe. 2006. Clinical decision-support systems. *Biomedical informatics*.
- António Palma-dos Reis, Fatemeh Zahedi, et al. 1999. Designing personalized intelligent financial decision support systems. *Decision Support Systems*, 26(1).
- Katharina Probst, Rayid Ghani, Marko Krema, Andy Fano, and Yan Liu. 2007. Extracting and using attribute-value pairs from product descriptions on the web. *From Web to Social Web: Discovering and Deploying User and Content Profiles*.
- Alexander I. Rudnicky, Christina L. Bennett, Alan W. Black, Ananlada Chotimongkol, Kevin A. Lenzo, Alice Oh, and Rita Singh. 2000. Task and domain specific modelling in the carnegie mellon communicator system. In *INTERSPEECH*.
- Hassan Sajjad, Patrick Pantel, and Micheal Gamon. 2012. Underspecified query refinement via natural language question generation. In *COLING*.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *NEMLP*.
- Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *ICIA*.
- Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen, and Lee Hetherington. 2000. Jupiter: A telephone-based conversational interface for weather information. *IEEE Trans. on Speech and Audio Processing*.



# Ensemble Triangulation for Statistical Machine Translation\*

Majid Razmara and Anoop Sarkar  
School of Computing Science  
Simon Fraser University  
Burnaby, BC, Canada  
{razmara,anoop}@sfu.ca

## Abstract

State-of-the-art statistical machine translation systems rely heavily on training data and insufficient training data usually results in poor translation quality. One solution to alleviate this problem is *triangulation*. Triangulation uses a third language as a pivot through which another source-target translation system can be built. In this paper, we dynamically create multiple such triangulated systems and combine them using a novel approach called *ensemble decoding*. Experimental results of this approach show significant improvements in the BLEU score over the direct source-target system. Our approach also outperforms a strong linear mixture baseline.

## 1 Introduction

The objective of current statistical machine translation (SMT) systems is to build cheap and rapid corpus-based SMT systems without involving human translation expertise. Such SMT systems rely heavily on their training data. State-of-the-art SMT systems automatically extract translation rules (e.g. phrase pairs), learn segmentation models, re-ordering models, etc. and find tuning weights solely from data and hence they rely heavily on high quality training data. There are many language pairs for which there is no parallel data or the available data is not sufficiently large to build a reliable SMT system. For example, there is no Chinese-Farsi parallel text, although there exists sufficient parallel data between these two languages and English. For SMT, an important research direction is to improve the quality of translation when there is no, insufficient or poor-quality parallel data between a pair of languages.

\*This research was partially supported by an NSERC, Canada (RGPIN: 264905) grant and a Google Faculty Award to the second author.

One approach that has been recently proposed is *triangulation*. Triangulation is the process of translating from a source language to a target language via an intermediate language (aka pivot, or bridge). This is very useful specifically for low-resource languages as SMT systems built using small parallel corpora perform poorly due to data sparsity. In addition, ambiguities in translating from one language into another may disappear if a translation into some other language is available.

One obvious benefit of triangulation is to increase the coverage of the model on the input text. In other words, we can reduce the number of out-of-vocabulary words (OOVs), which are a major cause of poor quality translations, using other paths to the target language. This can be especially helpful when the model is built using a small amount of parallel data.

Figure 1 shows how triangulation can be useful in reducing the number of OOVs when translating from French to English through three pivot languages: Spanish (*es*), German (*de*) and Italian (*it*). The solid lines show the number of OOVs for a direct MT system with regard to a multi-language parallel test set (Section 6.2 contains the details about the data sets) and the dotted lines show the OOVs in the triangulated ( $src \rightarrow pvt \rightarrow tgt$ ) systems. The number of OOVs on triangulated paths can never be less than the first edge (i.e.  $src \rightarrow pvt$ ) and it is usually higher than the second edge (i.e.  $pvt \rightarrow tgt$ ) as well. Thus, the choice of intermediate language is very important in triangulation.

Figure 1 also shows how combining multiple triangulated systems can reduce this number from 2600 (16%) OOVs to 1536 (9%) OOVs. Thus, combining triangulated systems with the original  $src \rightarrow tgt$  system is a good idea. When combining multiple systems, the upper bound on the number of OOVs is the minimum among all OOVs in the different triangulations. These OOV rates provide useful hints, among other clues, as to which pivot

languages will be more useful. In Figure 1, we can expect Italian (*it*) to help more than Spanish (*es*) and both to help more than German (*de*) in translation from French (*fr*) to English (*en*), which we confirmed in our experimental results (Table 1).

In addition to providing translations for otherwise untranslatable phrases, triangulation can find new translations for current phrases. The conditional distributions used for the translation model have been estimated on small amounts of data and hence are not robust due to data sparseness. Using triangulation, these distributions are smoothed and become more reliable as a result.

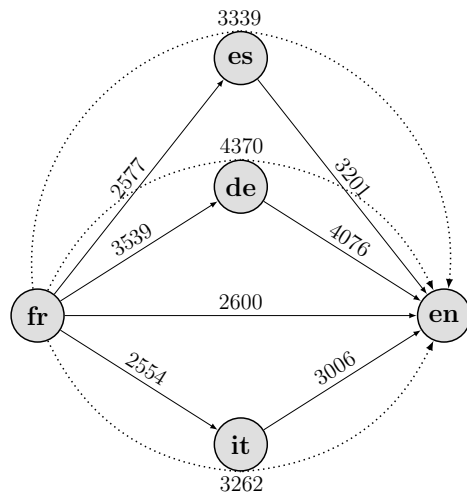
For each pivot language for which there exists parallel data with the source and the target language, we can create a  $src \rightarrow tgt$  system by bridging through the pivot language. If there are a number of such pivot languages with corresponding data, we can use mixture approaches to combine them in order to build a stronger model. We propose to apply the ensemble decoding approach of (Razmara et al., 2012) in this triangulation scenario. Ensemble decoding allows us to combine hypotheses from different models dynamically at the decoder. We experimented with 12 different language pairs and 3 pivot languages for each of them. Experimental results of this approach show significant improvements in the BLEU and METEOR scores over the direct source-target system in all the 12 language pairs. We also compare to a strong linear mixture baseline.

## 2 Related Work

Use of pivot languages in machine translation dates back to the early days of machine translation. Boitet (1988) discusses the choice of pivot languages, natural or artificial (e.g. interlingua), in machine translation. Schubert (1988) argues that a proper choice for an intermediate language for high-quality machine translation is a natural language due to the inherent lack of expressiveness in artificial languages. Previous work in applying pivot languages in machine translation can be categorized into these divisions:

### 2.1 System Cascades

In this approach, a  $src \rightarrow pvt$  translation system translates the source input into the pivot language and a second  $pvt \rightarrow tgt$  system takes the output of the previous system and translates it into the target language. Utiyama and Isahara (2007) use this



direct ( $fr-en$ )	2600 (16%)
triangulated ( $fr-\{es, de, it\}-en$ )	2066 (12%)
direct + triangulated	1536 (9%)

Figure 1: Number of OOVs when translating directly from *fr* to *en* (solid lines), triangulating through *es*, *de* or *it* individually (dotted lines), and when combining multiple triangulation systems with the direct system. OOV numbers are based on a multi-language parallel test set and the models are built on small corpora (10k sentence pairs), which are not multi-parallel.

approach to triangulate between Spanish, German and French through English. However, instead of using only the best translation, they took the  $n$ -best translations and translated them into the target language. MERT (Och, 2003) has been used to tune the weights for the new feature set which consists of  $src \rightarrow pvt$  and  $pvt \rightarrow tgt$  feature functions. The highest scoring sentence from the target language is used as the final translation. They showed that using 15 hypotheses in the  $pvt$  side is generally superior to using only one best hypothesis.

### 2.2 Corpus Synthesis

Given a  $pvt \rightarrow tgt$  MT system, one can translate the pivot side of a  $src-pvt$  parallel corpus into the target language and create a noisy  $src-tgt$  parallel corpus. This can also be exploited in the other direction, meaning that a  $pvt \rightarrow src$  MT system can be used to translate the pivot side of a  $pvt-tgt$  bitext. de Gispert and Marino (2006), for example, translated the Spanish side of an English-Spanish bitext into Catalan using an available Spanish-Catalan SMT system. Then, they built an English-Catalan MT system by training on this new parallel corpus.

### 2.3 Phrase-Table Triangulation

In this approach, instead of translating the input sentences from a source language to a pivot language and from that to a target language, triangulation is done on the phrase level by triangulating two phrase-tables:  $src \rightarrow pvt$  and  $pvt \rightarrow tgt$ :

$$(\bar{f}, \bar{e}) \in T_{\mathcal{F} \rightarrow \mathcal{E}} \iff \exists \bar{i} : (\bar{f}, \bar{i}) \in T_{\mathcal{F} \rightarrow \mathcal{I}} \wedge (\bar{i}, \bar{e}) \in T_{\mathcal{I} \rightarrow \mathcal{E}}$$

where  $\bar{f}, \bar{i}$  and  $\bar{e}$  are phrases in the source  $\mathcal{F}$ , pivot  $\mathcal{I}$  and target  $\mathcal{E}$  languages respectively and  $T$  is a set representing a phrase table.

Utiyama and Isahara (2007) also experimented with phrase-table triangulation. They compared both triangulation approaches when using Spanish, French and German as the source and target languages and English as the only pivot language. They showed that phrase-table triangulation is superior to the MT system cascades but both of them did not outperform the direct  $src \rightarrow tgt$  system.

The phrase-table triangulation approach with multiple pivot languages has been also investigated in several work (Cohn and Lapata, 2007; Wu and Wang, 2007). These triangulated phrase-tables are combined together using linear and log-linear mixture models. They also successfully combined the mixed phrase-table with a  $src-tgt$  phrase-table to achieve a higher BLEU score.

Bertoldi et al. (2008) formulated phrase triangulation in the decoder where they also consider the phrase-segmentation model between  $src-pvt$  and the reordering model between  $src-tgt$ .

Beside machine translation, the use of pivot languages has found applications in other NLP areas. Gollins and Sanderson (2001) used a similar idea in cross-lingual information retrieval where query terms were translated through multiple pivot languages to the target language and the translations are combined to reduce the error. Pivot languages have also been successfully used in inducing translation lexicons (Mann and Yarowsky, 2001) as well as word alignments for resource-poor languages (Kumar et al., 2007; Wang et al., 2006). Callison-Burch et al. (2006) used pivot languages to extract paraphrases for unknown words.

### 3 Baselines

In this paper, we compare our approach with two baselines. A simple baseline is the direct system

between the source and target languages which is trained on the same amount of parallel data as the triangulated ones. In addition, we implemented a phrase-table triangulation method (Cohn and Lapata, 2007; Wu and Wang, 2007; Utiyama and Isahara, 2007). This approach presents a probabilistic formulation for triangulation by marginalizing out the pivot phrases, and factorizing using the chain rule:

$$\begin{aligned} p(\bar{e} | \bar{f}) &= \sum_{\bar{i}} p(\bar{e}, \bar{i} | \bar{f}) \\ &= \sum_{\bar{i}} p(\bar{e} | \bar{i}, \bar{f}) p(\bar{i} | \bar{f}) \\ &\approx \sum_{\bar{i}} p(\bar{e} | \bar{i}) p(\bar{i} | \bar{f}) \end{aligned}$$

where  $\bar{f}, \bar{e}$  and  $\bar{i}$  are phrases in the source, target and intermediate language respectively. In this equation, a conditional independence assumption has been made that source  $\bar{f}$  and target phrases  $\bar{e}$  are independent given their corresponding pivot phrase(s)  $\bar{i}$ . The equation requires that all phrases in the  $src \rightarrow pvt$  direction must also appear in  $pvt \rightarrow tgt$ . All missing phrases are simply dropped from the final phrase-table.

Using this approach, a triangulated source-target phrase-table is generated for each pivot language. Then, linear and log-linear mixture methods are used to combine these phrase-tables into a single phrase-table in order to be used in the decoder. We implemented the linear mixture approach, since linear mixtures often outperform log-linear ones (Cohn and Lapata, 2007). We then compare the results of these baselines with our approach over multiple language pairs (Section 6.2). In linear mixture models, each feature in the mixture phrase-table is computed as a linear interpolation of corresponding features in the component phrase-tables using a weight vector  $\vec{\lambda}$ .

$$\begin{aligned} p(\bar{e} | \bar{f}) &= \sum_i \lambda_i p_i(\bar{e} | \bar{f}) \\ p(\bar{f} | \bar{e}) &= \sum_i \lambda_i p_i(\bar{f} | \bar{e}) \\ \forall \lambda_i > 1 \quad \sum_i \lambda_i &= 1 \end{aligned}$$

Following Cohn and Lapata (2007), we combined triangulated phrase-tables with uniform weights into a single phrase table and then interpolated it with the phrase-table of the direct system.

## 4 Ensemble Decoding

SMT log-linear models (Koehn, 2010) find the most likely target language output  $e$  given the source language input  $f$  using a vector of feature functions  $\phi$ :

$$p(e|f) \propto \exp(\mathbf{w} \cdot \phi)$$

Ensemble decoding combines several models dynamically at the decoding time. The scores are combined for each partial hypothesis using a user-defined mixture operation  $\odot$  over component models.

$$p(e|f) \propto \exp(\mathbf{w}_1 \cdot \phi_1 \odot \mathbf{w}_2 \cdot \phi_2 \odot \dots)$$

Razmara et al. (2012) successfully applied ensemble decoding to domain adaptation in SMT and showed that it performed better than approaches that pre-compute linear mixtures of different models. Several mixture operations were proposed, allowing the user to encode belief about the relative strengths of the component models. These mixture operations receive two or more probabilities and return the mixture probability  $p(\bar{e}|\bar{f})$  for each rule  $\bar{f} \rightarrow \bar{e}$  used in the decoder. Different options for these operations are:

- **Weighted Sum (wsum)** is defined as:

$$p(\bar{e}|\bar{f}) \propto \sum_m^M \lambda_m \exp(\mathbf{w}_m \cdot \phi_m)$$

where  $m$  denotes the index of component models,  $M$  is the total number of them and  $\lambda_m$  is the weight for component  $m$ .

- **Weighted Max (wmax)** is defined as:

$$p(\bar{e}|\bar{f}) \propto \max_m (\lambda_m \exp(\mathbf{w}_m \cdot \phi_m))$$

- **Model Switching (Switch)**: Each cell in the CKY chart is populated only by rules from one of the models and the other models' rules are discarded. Each component model is considered an expert on different spans of the source. A binary indicator function  $\delta(\bar{f}, m)$  picks a component model for each span:

$$\delta(\bar{f}, m) = \begin{cases} 1, & m = \operatorname{argmax}_{n \in M} \psi(\bar{f}, n) \\ 0, & \text{otherwise} \end{cases}$$

The criteria for choosing a model for each cell,  $\psi(\bar{f}, n)$ , is based on max top score, i.e.

for each cell, the model that has the highest weighted best-rule score wins:

$$\psi(\bar{f}, n) = \lambda_n \max_{\bar{e}} (\mathbf{w}_n \cdot \phi_n(\bar{e}, \bar{f}))$$

The probability of each phrase-pair  $(\bar{e}, \bar{f})$  is then:

$$p(\bar{e}|\bar{f}) = \sum_m^M \delta(\bar{f}, m) p_m(\bar{e}|\bar{f})$$

## 5 Our Approach

### 5.1 Dynamic Triangulation

Given a  $src \rightarrow pvt$  and a  $pvt \rightarrow tgt$  system which are independently trained and tuned on their corresponding parallel data, these two systems can be triangulated dynamically in the decoder.

For each source phrase  $\bar{f}$ , the decoder consults the  $src \rightarrow pvt$  system to get its translations on the pivot side  $\bar{i}$  with their scores. Consequently, each of these pivot-side translation phrases is queried from the  $pvt \rightarrow tgt$  system to obtain their translations on the target side with their corresponding scores. Finally a  $(\bar{f}, \bar{e})$  pair is constructed from each  $(\bar{f}, \bar{i})$  and  $(\bar{i}, \bar{e})$  pair, whose score is computed as:

$$p_{\mathcal{I}}(\bar{f}|\bar{e}) \propto \max_{\bar{i}} \exp \left( \underbrace{w_1 \cdot \phi_1(\bar{f}, \bar{i})}_{\mathcal{F} \rightarrow \mathcal{I}} + \underbrace{w_2 \cdot \phi_2(\bar{i}, \bar{e})}_{\mathcal{I} \rightarrow \mathcal{E}} \right)$$

This method requires the language model score of the  $src \rightarrow pvt$  system. However for simplicity we do not use the pivot-side language models and hence the score of the  $src \rightarrow pvt$  system does not include the language model and word penalty scores. In this formulation for a given source and target phrase pair  $(\bar{f}, \bar{e})$ , if there are multiple bridging pivot phrases  $\bar{i}$ , we only use the one that yields the highest score. This is in contrast with previous work where they take the sum over all such pivot phrases (Cohn and Lapata, 2007; Utiyama and Isahara, 2007). We use *max* as it outperforms *sum* in our preliminary experiments.

It is noteworthy that in computing the score for  $p_{\mathcal{I}}(\bar{f}|\bar{e})$ , the scores from  $src \rightarrow pvt$  and  $pvt \rightarrow tgt$  are added uniformly. However, there is no reason why this should be the case. Two different weights can be assigned to these two scores to highlight the importance of one against the other one.

A naive implementation of phrase-triangulation in the decoder would require  $O(n^2)$  steps for each source sub-span, where  $n$  is the average number of translation fan-out (i.e. possible translations) for each phrase. However, since the phrase candidates from both  $src \rightarrow pvt$  and  $pvt \rightarrow tgt$  are already sorted, we use a lazy algorithm that reduces the computational complexity to  $O(n)$ .

## 5.2 Combining Triangulated Systems

If we can make use of multiple pivot languages, a system can be created on-the-fly for each pivot language by triangulation and these systems can then be combined together in the decoder using *ensemble decoding* discussed in Section 4. Following previous work, these triangulated phrase-tables can also be combined with the direct system to produce a yet stronger model. However, we do not combine them in two steps. Instead, all triangulated systems and the direct one are combined together in a single step.

Ensemble decoding is aware of full model scores when it compares, ranks and prunes hypotheses. This includes the language model, word, phrase and glue rule penalty scores as well as standard phrase-table probabilities.

Since ensemble decoding combines the scores of common hypotheses across multiple systems rather than combining their feature values as in mixture models, it can be used to triangulate heterogeneous systems such as phrase-based, hierarchical phrase-based, and syntax-based with completely different feature types. Considering that ensemble decoding can be used in these diverse scenarios, it offers an attractive alternative to current phrase-table triangulation systems.

## 5.3 Tuning Component Weights

Component weights control the contribution of each model in the ensemble. A tuning procedure should assign higher weights to the models that produce higher quality translations and lower weights to weak models in order to control their noise propagation in the ensemble. In the ensemble decoder, since we do not have explicit gradient information for the objective function, we use a direct optimizer for tuning. We used *Condor* (Vanden Berghen and Bersini, 2005) which is a publicly available toolkit based on Powell’s algorithm.

The ensemble between three triangulated models and a direct one requires tuning in a 4-

dimensional space, one for each system. If, on average, the tuner evaluates the decoder  $n$  times in each direction in the optimization space, there needs to be  $n^4$  ensemble decoder evaluations, which is very time consuming. Instead, we resorted to a simpler approach for tuning: each triangulated model is separately tuned against the direct model with a fixed weights (we used a weight of 1). In other words, three ensemble models are created, each on a single triangulated model plus the direct one. These ensembles are separately tuned and once completed, these weights comprise the final tuned weights. Thus, the total number of ensemble evaluations reduces from  $O(n^4)$  to  $O(3n)$ .

In addition to this significant complexity reduction, this method enables parallelism in tuning, since the three individual tuning branches can now be run independently. The final tuned weights are not necessarily a local optima and one can run further optimization steps around this point to get to even better solutions which should lead to higher BLEU scores.

# 6 Experiments & Results

## 6.1 Experimental Setup

For our experiments, we used the Europarl corpus (v7) (Koehn, 2005) for training sets and ACL/WMT 2005<sup>1</sup> data for dev/test sets (2k sentence pairs) following Cohn and Lapata (2007). Our goal in this paper was to understand how multiple languages can help in triangulation, the improvement in coverage of the unseen data due to triangulation, and the importance of choosing the right languages as pivot languages. Thus, we needed to run experiments on a large number of language pairs, and for each language pair we wanted to work with many pivot languages. To this end, we created small sub-corpora from Europarl by sampling 10,000 sentence pairs and conducted our experiments on them. As we will show, using larger data than this would result in prohibitively large triangulated phrase tables. Table 2 shows the number of words on both sides of used language pairs in our corpora.

The ensemble decoder is built on top of an in-house implementation of a Hiero-style MT system (Chiang, 2005) called Kriya (Sankaran et al., 2012). This Hiero decoder obtains BLEU

<sup>1</sup><http://www.statmt.org/wpt05/mt-shared-task/>

src↓		tgt →		en	es	fr
de	pivots	en	–	15.94	13.62	
		es	14.47	–	13.43	
		fr	14.39	13.45	–	
		it	14.14	14.90	11.67	
	direct	21.94	20.70	17.37		
	mixture	21.86	<b>22.30</b>	<b>18.28</b>		
	wmax	22.49	21.32	18.22		
	wsum	22.22	21.42	17.98		
	switch	<b>22.59</b>	21.80	17.70		

src↓		tgt →		de	es	fr
en	pivots	de	–	20.47	17.38	
		es	12.95	–	20.78	
		fr	14.09	23.25	–	
		it	13.00	23.18	19.02	
	direct	17.57	28.81	24.58		
	mixture	<b>17.91</b>	28.89	24.30		
	wmax	17.77	29.17	<b>25.39</b>		
	wsum	17.68	<b>29.33</b>	24.70		
	switch	17.77	29.32	24.98		

src↓		tgt →		de	en	fr
es	pivots	de	–	18.84	23.28	
		en	14.50	–	18.55	
		fr	12.48	22.81	–	
		it	13.69	23.14	23.44	
	direct	16.30	28.11	29.83		
	mixture	<b>17.75</b>	28.99	29.47		
	wmax	17.34	<b>29.23</b>	<b>30.54</b>		
	wsum	16.79	28.79	30.12		
	switch	16.53	29.16	29.68		

src↓		tgt →		de	en	es
fr	pivots	de	–	20.15	22.96	
		en	14.84	–	27.84	
		es	14.35	23.59	–	
		it	14.08	24.08	30.38	
	direct	16.56	28.79	35.27		
	mixture	17.39	28.83	35.27		
	wmax	<b>17.67</b>	<b>29.95</b>	<b>36.07</b>		
	wsum	17.41	28.62	35.98		
	switch	17.78	28.79	36.33		

Table 1: Results of i) single-pivot triangulation; ii) baseline systems including direct systems and linear mixture of triangulated phrase-tables; iii) ensemble triangulation results based on different mixture operations. The mixture and ensemble methods are based on multi-pivot triangulation. These methods are built on 10k sentence-pair corpora.

$L_1 - L_2$	$L_1$ tokens (K)	$L_2$ tokens (K)
de - en	232	249
de - es	232	263
de - fr	231	259
de - it	245	253
en - es	250	264
en - fr	251	262
en - it	260	251
es - fr	262	261
es - it	274	252
fr - it	272	251

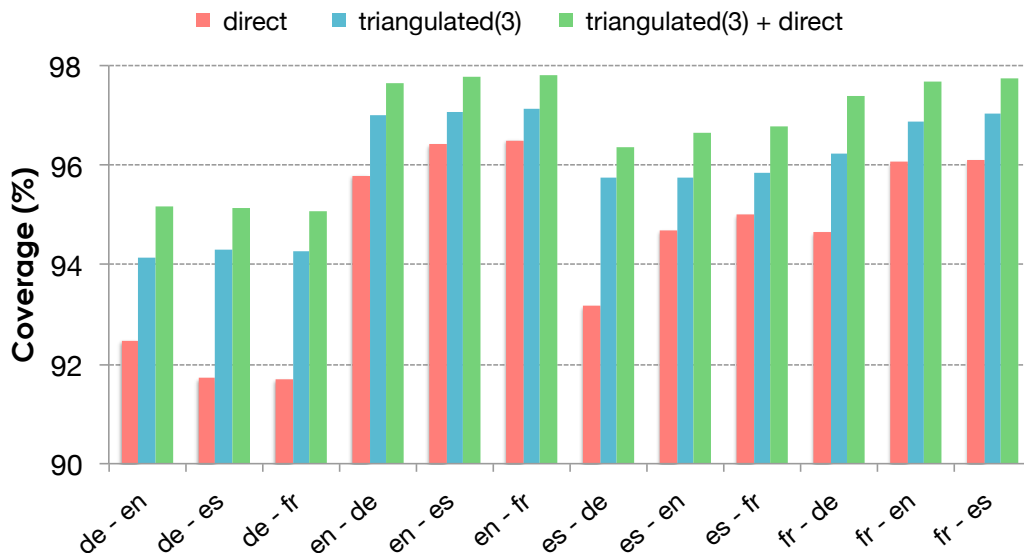
Table 2: Number of tokens in each language pair in the training data.

scores equal to or better than the state-of-the-art in phrase-based and hierarchical phrase-based translation over a wide variety of language pairs and data sets. It uses the following standard features: forward and backward relative-frequency and lexical TM probabilities; LM; word, phrase and glue-rules penalty. GIZA++ (Och and Ney,

2000) has been used for word alignment with phrase length limit of 10. In both systems, feature weights were optimized using MERT (Och, 2003). We used the target sides of the Europarl corpus (2M sentences) to build 5-gram language models and smooth them using the Kneser-Ney method. We used SRILM (Stolcke, 2002) as the language model toolkit.

## 6.2 Results

Table 1 shows the BLEU scores when using two languages from  $\{fr, en, es, de\}$  as source and target, and the other two languages plus *it* as intermediate languages. The first group of numbers are BLEU scores for triangulated systems through the specified pivot language. For example, translating from *de* to *es* through *en* (i.e.  $de \rightarrow en \rightarrow es$ ) gets 15.94% BLEU score. The second group shows the BLEU scores of the baseline systems including the direct system between the source and target languages and the linear mixture baseline of the three triangulated systems. The BLEU scores of ensemble decoding using different mixture op-



direct	478K	393K	403K	665K	1,084K	1,155K	479K	927K	1,319K	394K	743K	976K
tri + direct	83M	102M	132M	113M	103M	133M	129M	101M	152M	141M	109M	129M

Figure 2: Coverage for i) direct system; ii) combined triangulated system with three 3 languages; and iii) the combination of the triangulated phrase-tables and the direct one. The table shows the number of rules for each system and language pair after filtering based on the source side of the test set.

erations are illustrated at the bottom.

As the table shows, our approach outperforms the direct systems in all the 12 language pairs while the mixture model systems fail to improve over the direct system baseline for some of the language pairs. Our approach also outperforms the mixture models in most cases. Overall, ensemble decoding with *wmax* as mixture operation performs the best among the different systems and baselines. Figure 3 shows the average of the BLEU score of the direct system, mixture models and *wmax* on all 12 systems. On average the *wmax* method obtains 0.33 BLEU points higher than the mixture models.

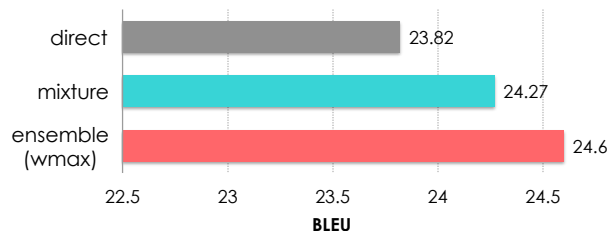


Figure 3: The average BLEU scores of the direct system, mixture models and *wmax* ensemble triangulation approach over all 12 language pairs.

We also computed the Meteor scores (Denkowski and Lavie, 2011) for all systems

and the results are summarized in Figure 4. As the figure illustrates, our ensemble decoding approach with *wmax* outperforms the mixture models in 11 of 12 language pairs based on Meteor scores.

### 6.3 Phrase table coverage

Figure 2 shows the phrase-table coverage of the test set for different language pairs. The coverage is defined as the percentage of unigrams in the source side of the test set for which the corresponding phrase-table has translations for. The first set of bars shows the coverage of the direct systems and the second one shows that of the combined triangulated systems for three pivot languages. Finally, the last set of bars indicate the coverage when the direct phrase-table is combined with the triangulated ones. In all language pairs, the combined triangulated phrase-tables have a higher coverage compared to the direct phrase-tables. As expected, the coverage increases when these two phrase-tables are aggregated. The table below the figure shows the number of rules for each system and language pair after filtering out based on the source side of the test set. This illustrates why running experiments on larger sizes of parallel data is prohibitive for hierarchical phrase-based models.

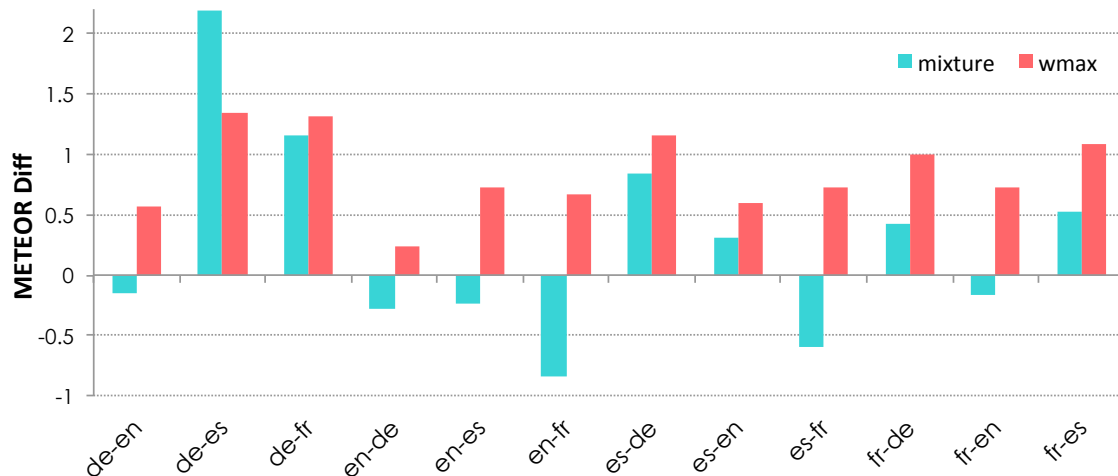


Figure 4: Meteor score difference between mixture models and direct systems as well as the difference between ensemble decoding approach with *wmax* and the direct system.

### 6.3.1 Choice of Pivot Language

Cohn and Lapata (2007) showed that the pivot language should be close to the source or the target language in order to be effective. For example, when translating between Romance languages (Italian, Spanish, etc.), the pivot language should also be a Romance language. In addition to those findings, based on the results presented in Table 1, here are some observations for these five European languages:

- When translating from or to *de*, *en* is the best pivot language;
- Generally *de* is not a suitable pivot language for any translation pair;
- When translating from *en* to any other language, *fr* is the best pivot;
- *it* is the best intermediate language when translating from *fr* or *es* to other languages; except when translating to *de* for which *en* is the best pivot language (c.f. first finding);

## 7 Conclusion and Future Work

In the paper, we introduced a novel approach for triangulation which does phrase-table triangulation and model combination on-the-fly in the decoder. Ensemble decoder uses the full hypothesis score for triangulation and combination and hence is able to mix hypotheses from heterogeneous systems.

Another advantage of this method to the phrase-table triangulation approach is that our method is

applicable even when there exists no parallel data between source and target languages for tuning because we only use the *src-tgt* tuning set to optimize hyper-parameters, though phrase-table triangulation methods use it to learn MT log-linear feature weights for which having a tuning set is much more essential. Empirical results also showed that this method with *wmax* outperforms the baselines.

Future work includes imposing restrictions on the generated triangulated rules in order to keep only ones that have a strong support from the word alignments. By exploiting such constraints, we can experiment with larger sizes of parallel data. Specifically, a more natural experimental setup for triangulation which we would like to try is to use a small direct system with big *src*  $\rightarrow$  *pvt* and *pvt*  $\rightarrow$  *tgt* systems. This resembles the actual situation for resource-poor language pairs. We will also experiment with higher number of pivot languages.

Currently, most research in this area focuses on triangulation on paths containing only one pivot language. We can also analyze our method when using more languages in the triangulation chain and see whether there would any gain in doing such.

Finally, in current methods all  $(\bar{f}, \bar{i})$  phrase pairs of the *src*  $\rightarrow$  *pvt* systems, for which there does not exist any  $(\bar{i}, \bar{e})$  pair in *pvt*  $\rightarrow$  *tgt* are simply discarded. However in most cases, such  $\bar{i}$  phrases can be segmented into smaller phrases (or rules for Hero systems) to be triangulated via them. This segmentation is a decoding problem which requires an efficient algorithm to be practical.



## References

- N. Bertoldi, M. Barbaiani, M. Federico, and R. Cattoni. 2008. Phrase-based statistical machine translation with pivot languages. *Proceeding of IWSLT*, pages 143–149.
- C. Boitet. 1988. Pros and cons of the pivot and transfer approaches in multilingual machine translation. *Maxwell et al.(1988)*, pages 93–106.
- C. Callison-Burch, P. Koehn, and M. Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24. Association for Computational Linguistics.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA. ACL.
- Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic, June. Association for Computational Linguistics.
- A. de Gispert and J.B. Marino. 2006. Catalan-english statistical machine translation without parallel corpus: bridging through spanish. In *Proc. of 5th International Conference on Language Resources and Evaluation (LREC)*, pages 65–68.
- Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.
- T. Gollins and M. Sanderson. 2001. Improving cross language retrieval with triangulated translation. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 90–95. ACM.
- Martin Kay. 1997. The proper place of men and machines in language translation. *Machine Translation*, 12(1/2):3–23.
- P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.
- Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.
- Shankar Kumar, Franz Josef Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *EMNLP-CoNLL*, pages 42–50. ACL.
- Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA.
- F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the ACL*, pages 440–447, Hongkong, China, October.
- Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41th Annual Meeting of the ACL*, Sapporo, July. ACL.
- Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 940–949. The Association for Computer Linguistics.
- Baskaran Sankaran, Majid Razmara, and Anoop Sarkar. 2012. Kriya – an end-to-end hierarchical phrase-based mt system. *The Prague Bulletin of Mathematical Linguistics*, 97(97), April.
- K. Schubert. 1988. Implicitness as a guiding principle in machine translation. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 599–601. Association for Computational Linguistics.
- Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286.
- M. Utiyama and H. Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Proceedings of NAACL-HLT*, volume 7, pages 484–491.
- Frank Vanden Berghen 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.
- H. Wang, H. Wu, and Z. Liu. 2006. Word alignment for languages with scarce resources using bilingual corpora of other language pairs. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 874–881. Association for Computational Linguistics.
- H. Wu and H. Wang. 2007. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181.

# Robust Transliteration Mining from Comparable Corpora with Bilingual Topic Models

John Richardson<sup>†</sup>

Toshiaki Nakazawa<sup>‡</sup>

Sadao Kurohashi<sup>†</sup>

<sup>†</sup>Graduate School of Informatics, Kyoto University, Kyoto 606-8501

<sup>‡</sup>Japan Science and Technology Agency, Kawaguchi-shi, Saitama 332-0012

john@nlp.ist.i.kyoto-u.ac.jp, nakazawa@pa.jst.jp, kuro@i.kyoto-u.ac.jp

## Abstract

We present a high-precision, language-independent transliteration framework applicable to bilingual lexicon extraction. Our approach is to employ a bilingual topic model to enhance the output of a state-of-the-art grapheme-based transliteration baseline. We demonstrate that this method is able to extract a high-quality bilingual lexicon from a comparable corpus, and we extend the topic model to propose a solution to the out-of-domain problem.

## 1 Introduction

A large, high-quality bilingual lexicon is of great utility to any dictionary-based system that processes bilingual data. The ability to automatically generate such a lexicon without relying on expensive training data or pre-existing lexical resources allows us to find translations for rare and unknown words with high efficiency.

Transliteration<sup>1</sup> is particularly important as new words are often created by importing words from other languages, especially English. It would be an almost impossible task to create and maintain a dictionary of such words by hand, as new words appear rapidly, especially in online texts, and word usage can vary over time.

In this paper we construct a language-independent transliteration framework. Our model builds on previous transliteration work, improving extraction and generation precision by including semantic as well as purely lexical features. The proposed model can be trained

<sup>1</sup>This paper considers both ‘transliteration’ (EN–XX) and ‘back-transliteration’ (XX–EN). For simplicity we refer to both tasks as ‘transliteration’.

on comparable corpora, thereby not relying on expensive or often unavailable parallel data.

The motivation behind the approach of combining lexical and semantic features is that these two components are largely independent, greatly improving the effectiveness of their combination. This is particularly important for word-sense disambiguation. For example, a purely lexical approach is not sufficient to transliterate the Japanese ソース (*soosu*), as it can mean either ‘sauce’ or ‘source’ depending on the context.

## 2 Previous Work

Previous work has considered various methods for transliteration, ranging from simple edit distance and noisy-channel models (Brill et al., 2001) to conditional random fields (Ganesh et al., 2008) and finite state automata (Noeman and Madkour, 2010). We construct a baseline by modelling transliteration as a Phrase-Based Statistical Machine Translation (PB-SMT) task, a popular and well-studied approach (Matthews, 2007; Hong et al., 2009; Antony et al., 2010).

The vast majority of previous work on transliteration has considered only lexical features, for example spelling similarity and transliteration symbol mapping, however we build on the inspiration of Li et al. (2007) and later Hagiwara and Sekine (2012), who introduced semantic features to a transliteration model.

Li et al. (2007) proposed the concept of ‘semantic transliteration’, which is the consideration of inherent semantic information in transliterations. Their example is the influence of the source language and gender of foreign names on their transliterations into Chinese. Hagiwara and Sekine (2012) expanded upon this idea by considering a ‘latent class’

transliteration model considering transliterations to be grouped into categories, such as language of origin, which can give additional information about their formation. For example, if we know that a transliteration is of Italian origin, we are more likely to recover the letter sequence ‘gli’ than if it were originally French.

While these methods consider limited semantic features, they do not make use of the rich contextual information available from comparable corpora. We show such contextual information, in the form of bilingual topic distributions, to be highly effective in generating transliterations.

Bilingual lexicon mining from non-parallel data has been tackled in recent research such as Tamura et al. (2012) and Haghghi et al. (2008), and we build upon the techniques of multilingual topic extraction from Wikipedia pioneered by Ni et al. (2009). Previous research in lexicon mining has tended to focus on semantic features, such as context similarity vectors and topic models, but these have yet to be applied to the task of transliteration mining. We use the word-topic distribution similarities explored in Vulić et al. (2011) as baseline word similarity measures.

In some cases it is possible to use monolingual corpora for transliteration mining, as English is often written alongside transliterations (Kaji et al., 2011), however we consider the more general setting where such information is unavailable.

### 3 Baseline Transliteration Model

We begin by constructing a baseline transliteration system trained only on lexical features. This baseline system will allow us to compare directly the effectiveness of the addition of a semantic model to a traditional transliteration framework.

Our baseline model is a grapheme-based machine transliteration system. We model transliteration as a machine translation task on a character rather than word level, treating character groups as phrases. The model is trained by learning phrase alignments such as that shown in Figure 1. The field of phrase-based SMT has been well studied and there exists a standard toolset enabling the construc-

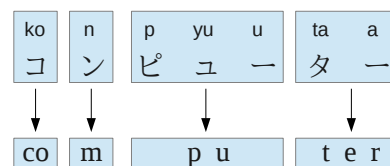


Figure 1: Example of Japanese–English transliteration phrase alignment.

tion of an easily reproducible baseline system.

We use the default configuration of Moses (Koehn et al., 2007) to train our baseline system, with the distortion limit set to 1 (as transliteration requires monotonic alignment). Character alignment is performed by GIZA++ (Och and Ney, 2003) with the ‘grow-diag-final’ heuristic for training. We apply standard tuning with MERT (Och, 2003) on the BLEU (Papineni et al., 2001) score. The language model is built with SRILM (Stolcke, 2002) using Kneser-Ney smoothing (Kneser and Ney, 1995).

The system described above has been implemented as specified in previous work such as Matthews (2007) (Chinese and Arabic), Hong et al. (2009) (Korean), and Antony et al. (2010) (Kannada). We demonstrate that this standard, highly-regarded baseline can be greatly improved with our proposed method.

## 4 Semantic Model

Having set up the baseline system, we turn to the task of combining a semantic model with our transliteration engine. We employ the method of bilingual LDA (Mimno et al., 2009), an extension of monolingual Latent Dirichlet Allocation (LDA) (Blei et al., 2003) as the semantic model.

Monolingual LDA takes as its input a set of monolingual documents and generates a word-topic distribution  $\phi$  classifying words appearing in these documents into semantically similar topics. Bilingual LDA extends this by considering pairs of comparable documents in each of two languages, and outputs a pair of word-topic distributions  $\phi$  and  $\psi$ , one for each input language. The graphical model for bilingual LDA is illustrated in Figure 2.

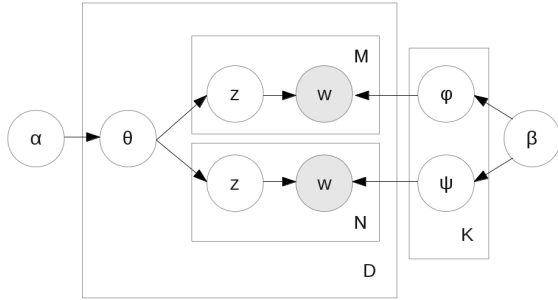


Figure 2: Graphical model for Bilingual LDA with  $K$  topics,  $D$  document pairs and hyperparameters  $\alpha$  and  $\beta$ . Topics for each document are sampled from the common distribution  $\theta$ , and the two languages have word-topic distributions  $\phi$  and  $\psi$ .

#### 4.1 Motivation for Bilingual LDA

We choose to employ a bilingual topic model to measure semantic similarity (i.e. topic similarity) of word pairs rather than the more intuitive method of comparing monolingual context similarity vectors (Rapp, 1995) for reasons of robustness and scalability.

Measuring context similarity on a word level requires a bilingual lexicon to match cross-language word pairs and such bilingual data is often expensive or unavailable. There are also problems with directly comparing collocations and word concurrence of distant language pairs as they do not always correspond predictably. Therefore our proposed method provides a more robust approach using coarser semantic features.

The use of topic models as a semantic similarity measure is a scalable method because document-aligned bilingual training data is growing ever more widely available. Examples of such sources are Wikipedia, multilingual newspaper articles and mined Web data.

#### 4.2 Semantic Similarity Measures

In order to apply bilingual topic models to a transliteration task, we must construct an effective word similarity measure for source and target transliteration candidates. We improve upon three natural similarity measures, Cos, Cue and KL, based on those considered in Vulić et al. (2011), by proposing two methods of feature combination: reordering and SVM

combination.

The reranking method considers hybrid scores Base+Cos, Base+Cue and Base+KL. These are generated by reranking the top-10 baseline (Base) transliteration candidates by their respective semantic scores (Cos, Cue or KL). We used 10 candidates for filtering as we found this gave the best balance between volume and accuracy in preliminary experiments. Approximately 75–85% of correct transliterations (depending on language pair) were within the top-10 candidates and this is therefore an upper bound for the hybrid model accuracy. As a comparison, the top-100 candidates contained roughly 80–85% of correct transliterations, the remainder failing to be identified by the baseline.

We additionally consider the combination of all three semantic features with the baseline (Moses) transliteration scores using a Support Vector Machine (SVM) (Vapnik, 1995). The SVM is used to classify candidate pairs as ‘transliteration’ (positive) or ‘not transliteration’ (negative), and we rerank the candidates by SVM predicted values. The features used for SVM training are baseline, Cos, Cue and KL scores.

The similarity measures Cos, Cue and KL are defined below.

##### 4.2.1 Cos Similarity

The **Cos** method calculates the cosine similarity of the topic distribution vectors  $\psi_{k,w_e}$  and  $\phi_{k,w_f}$  for transliteration pair candidates  $w_e$  and  $w_f$ .

$$\text{Cos}(w_e, w_f) = \frac{\sum_{k=1}^K \psi_{k,w_e} \phi_{k,w_f}}{\sqrt{\sum_{k=1}^K \psi_{k,w_e}^2} \sqrt{\sum_{k=1}^K \phi_{k,w_f}^2}} \quad (1)$$

##### 4.2.2 Cue Similarity

The **Cue** method expresses the mean of the two probabilities  $P(w_e | w_f)$  of a transliteration  $w_e$  given some source language string  $w_f$  and  $P(w_f | w_e)$  of the reverse. We define:

$$P(w_e | w_f) = \sum_{k=1}^K \psi_{k,w_e} \frac{\phi_{k,w_f}}{\text{Norm}_\phi}$$

and likewise for  $P(w_e | w_f)$ , with the

normalization factors given by  $Norm_\phi = \sum_{k=1}^K \phi_{k,w_f}$  and  $Norm_\psi = \sum_{k=1}^K \psi_{k,w_e}$ .

Finally, we consider:

$$Cue(w_e, w_f) = \frac{1}{2}(P(w_e | w_f) + P(w_f | w_e)) \quad (2)$$

### 4.2.3 KL Similarity

The **KL** method considers the averaged Kullback-Leibler divergence:

$$KL(w_e, w_f) = \frac{1}{2}(KL_{e,f} + KL_{f,e}) \quad (3)$$

$$KL_{e,f} = \sum_{k=1}^K \frac{\phi_{k,w_e}}{Norm_\phi} \log \frac{\phi_{k,w_e}/Norm_\phi}{\psi_{k,w_f}/Norm_\psi}$$

$$KL_{f,e} = \sum_{k=1}^K \frac{\psi_{k,w_f}}{Norm_\psi} \log \frac{\psi_{k,w_f}/Norm_\psi}{\phi_{k,w_e}/Norm_\phi}$$

using the same normalization factors as for Cue similarity.

## 5 Experiments

In order to demonstrate the effectiveness of our proposed model, we constructed an evaluation framework for a transliteration extraction task. The language pairs English–Japanese (EN–JA), Japanese–English (JA–EN), English–Korean (EN–KO) and Korean–English (KO–EN) were chosen to verify that this method is effective for a variety of languages and in both transliteration directions. Indeed, the methods introduced in this paper could also be applied directly to other languages with many transliterations, such as Chinese, Arabic and Hindi.

While it is possible to make language-specific optimizations, we decided only to pre-process the data minimally (such as removing punctuation) in order to demonstrate that our model works effectively in a language-independent setting. Examples of language-specific preprocessing techniques that we did not perform include segmentation of Japanese compound nouns (Nakazawa et al., 2005) and splitting of Korean syllabic blocks (*eumjeols*) into smaller components (*jamo*) (Hong et al., 2009).

Language Pairs	Train	Tune	Test
EN–JA/JA–EN	59K	1K	1K
KO–EN/EN–KO	21K	1K	1K

Table 1: Number of aligned word pairs in each fold of data.

## 5.1 Data Set

We chose to build our data set from Wikipedia articles, as they provide document-aligned comparable data across a variety of languages. Figure 3 shows how the Wikipedia data was split.

### 5.1.1 Baseline Training Data

We trained our baseline system on aligned Wikipedia page titles. This data consisted of pairs of English and Japanese/Korean words extracted from the freely available Wikipedia XML dumps. The aligned titles were filtered with hand-written rules<sup>2</sup> to extract only transliteration pairs, and the test data was verified for correctness by hand. This data will be made available to encourage comparison for future transliteration research<sup>3</sup>.

The composition of this data is shown in Table 1. Aligned word pairs were shuffled randomly before splitting into the three folds to ensure an even topic distribution across each of ‘Train’, ‘Tune’ and ‘Test’.

### 5.1.2 Bilingual Topic Model

The bilingual topic model was trained on the body text of Wikipedia articles aligned with Wikipedia inter-language links. These correspond to articles covering the same content, however they are rarely of similar length and not necessarily close transliterations.

We first pre-processed the most recent Wikipedia XML dumps to remove all tags and data other than plain text sentences, then aligned articles with language links to generate comparable document pairs. Words occurring fewer than 10 or more than 100K times were also removed to reduce noise and computation time.

<sup>2</sup>Heuristic rules included extraction of Japanese katakana, a script used primarily for transliterations, and words aligned with proper nouns as defined in a name dictionary.

<sup>3</sup><http://orchid.kuee.kyoto-u.ac.jp/~john>

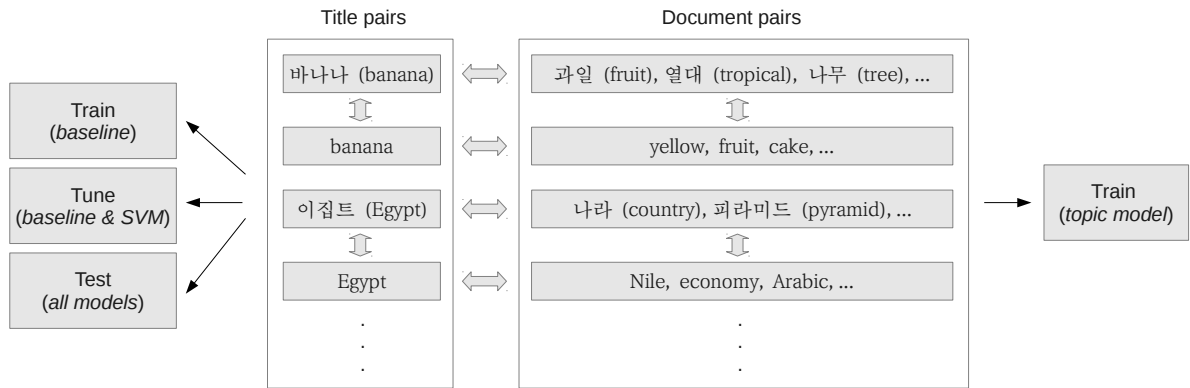


Figure 3: We extracted aligned title pairs (only transliterations) and aligned document pairs from Wikipedia using inter-language links. The baseline was trained and tuned on title pairs (‘Train’ and ‘Tune’), the topic model was trained on document pairs and the SVM was trained on the title pairs ‘Tune’ fold.

### 5.1.3 SVM Hybrid Model

The training data for the proposed SVM hybrid model was built from the same data used for the baseline (tuning fold). We first generated the top-10 distinct transliteration candidates for the tuning data using the ‘n-best-list’ option in Moses. These candidates were then labeled as ‘transliteration’ or ‘not-transliteration’ and feature scores (Base, Cos, Cue, KL) were generated for each candidate. The SVM model was trained using these labels and feature scores.

## 5.2 LDA Implementation Details

PolyLDA++, our implementation of multilingual LDA, was based on GibbsLDA++ (Phan et al., 2007), a toolkit for monolingual LDA. This software is available for free<sup>4</sup>.

Each topic model was trained over 1000 iterations, and the standard Dirichlet prior hyperparameters for the LDA model were set as  $\alpha = 50/K$  for  $K$  topics and  $\beta = 0.1$ .

The choice of number of topics is important, as demonstrated in Figure 4, which shows the top-1 accuracy of the SVM hybrid model using various numbers of topics  $K$ . The optimal value of  $K$  seems to be between around 100 for this data.

The model accuracy gradually decreases with adding more than 100 topics. We believe that this is because the granularity of

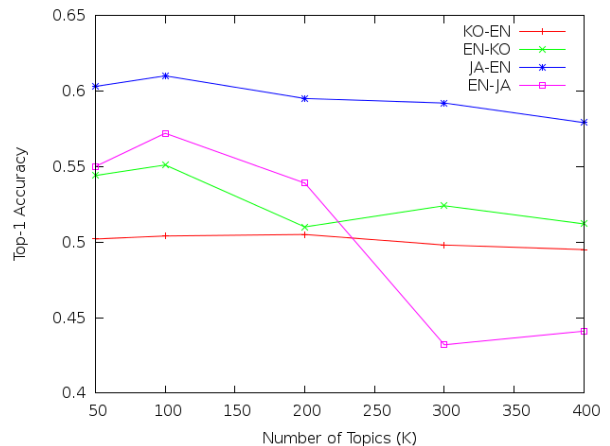


Figure 4: Top-1 accuracy of SVM for various  $K$ .

the topics becomes too fine to accommodate for the wide differences in semantic usage of English and Japanese/Korean transliteration pairs. A higher number of topics could be more suitable for more closely related language pairs, such as Italian and English (Vulić et al., 2011), because the higher similarity of word usage would allow for topics of more limited semantic scope. Such experiments are to be considered in future work. The results below are for  $K = 100$ .

## 5.3 Results

Table 2 compares the top-1 accuracy of our proposed hybrid models to the baseline perfor-

<sup>4</sup><http://orchid.kuee.kyoto-u.ac.jp/~john>

	JA-EN	EN-JA	KO-EN	EN-KO
Base	0.334	0.363	0.296	0.421
Base+Cos	0.608	0.559	0.494	0.516
Base+Cue	0.608	0.551	<b>0.507</b>	0.504
Base+KL	0.365	0.398	0.261	0.373
SVM	<b>0.610</b>	<b>0.572</b>	0.504	<b>0.551</b>

Table 2: Top-1 accuracy of proposed model for each hybrid scoring method.

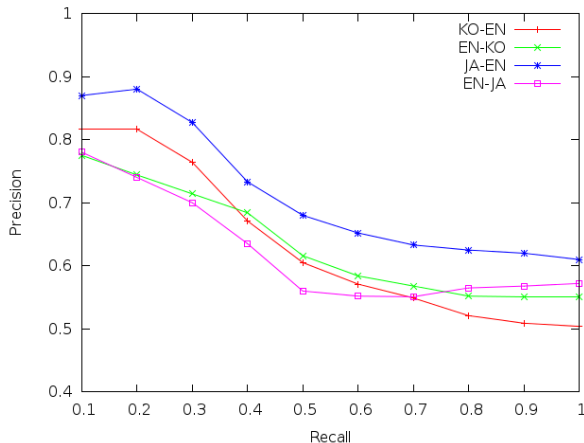


Figure 5: Precision-recall curve for SVM hybrid model.

mance. The SVM hybrid model outperformed the baseline for every language pair, by as much as 0.276 for JA-EN. This suggests that the addition of a bilingual topic model significantly improves transliteration accuracy.

In general the SVM was the most effective hybrid score, outperforming Base+Cos, Base+Cue and Base+KL in all but KO-EN, where it performed very slightly worse than Base+Cue.

Figure 5 shows the precision-recall curve for the SVM hybrid model over the test set. We vary recall by ranking the hybrid model scores for all test pairs and selecting only the highest scoring fraction to evaluate. This simulates a lexicon extraction task where we wish to sacrifice recall for precision. The results demonstrate that it is possible to improve significantly the precision of a set of extracted transliterations by reducing the recall. This large improvement is made possible because the topic similarity scores are particularly effective at measuring confidence in each transliteration candidate, allowing effective selection of the correct transliterations.

## 5.4 Comparison with Previous Work

The results compare favorably to the top-1 accuracy of similar existing systems, such as DIRECTL+ (Jiampojarn et al., 2010), which also used Wikipedia titles (EN-JA 0.398), and Hagiwara and Sekine (2012) (EN-JA 0.349).

Our baseline transliteration system can be measured against previous work using Moses and GIZA++ alignment, such as Matthews (2007) (EN-AR 0.43, AR-EN 0.39, EN-ZH 0.38, ZH-EN 0.35) and Hong et al. (2009) (EN-KO 0.45). These scores are consistent with our baseline results.

While it is difficult to compare directly the accuracy of transliteration systems across different languages and data sets, especially since we use additional data to train the semantic model, the results above show that our model has made a considerable improvement over the state-of-the-art baseline.

## 6 Extension to Out-of-Domain Words

The model described in this paper revolves around the use of a bilingual topic model to improve transliteration quality. What happens then when a source word is not covered by the topic model? This is a very important problem in a practical setting, and we show that even in such cases our model can improve considerably upon the baseline system. We define ‘out-of-domain’ words as source language words that did not appear in the topic model training data and hence do not have a known topic distribution.

### 6.1 Model Details

Our proposed approach is to consider not the word-topic distribution of the source word  $w_e$  itself, but rather that of the words in the surrounding context. We consider two methods for calculating the modified topic similarity

scores over the set of words  $W_e$  in the same context as the source word.

Let  $S(w_e, w_f)$  be a basic topic similarity score Cos, Cue or KL, then we define the extended scores  $ExtMean(W_e, w_f)$  and  $ExtWeight(W_e, w_f)$  as follows:

$$ExtMean(W_e, w_f) = \frac{\sum_{w_e \in W_e} S(w_e, w_f)}{|W_e|} \quad (4)$$

$$ExtWeight(W_e, w_f) = \frac{\sum_{w_e \in W_e} c'_{w_e} S(w_e, w_f)}{\sum_{w_e \in W_e} c'_{w_e}} \quad (5)$$

where  $c'_{w_e} = (\log c_{w_e})^{-1}$  for the frequency  $c_{w_e}$  of  $w_e$  appearing in the semantic model training data.

ExtMean corresponds to the mean topic similarity for each word in the context  $W_e$ . ExtWeight is weighted by the inverse log frequency of each word, allowing consideration of their semantic importance. These extended scores are used to train the SVM in place of the original scores.

## 6.2 Out-of-Domain Experiment

We performed an additional experiment where we transliterated a set of 25 Japanese words unknown to the topic model into English. These words appeared in Wikipedia fewer than 10 times and therefore were not included in our training data. We extracted the sentences and documents in which these words occurred, and back-transliterated the Japanese words into English by hand. We considered both sentence-level and document-level contexts for  $W_e$ , and evaluated each extended metric ExtMean and ExtWeight.

The results of the out-of-domain experiment are shown in Table 3, which gives the top-1 accuracy of the SVM hybrid model trained on the ExtMean and ExtWeight counterparts of Cos, Cue and KL similarities. Base is the top-1 accuracy using only the Moses baseline.

The most effective settings were to use ExtWeight on a sentence level context. There is a balance between size and relevance of context, with document-level context containing too many misleading words. The improvement of ExtWeight over ExtMean shows the impor-

	Base	ExtMean	ExtWeight
Document	0.27	0.44	0.48
Sentence		0.48	0.52

Table 3: Top-1 accuracy for out-of-domain model extension (JA-EN).

tance of weighting contextual words based on their importance (i.e. inverse log frequency).

The results show a large improvement (+0.25) over the baseline scores that is comparable to that of the in-domain model (+0.28, see Table 2). This suggests that the proposed model is an effective solution to the out-of-domain problem.

## 7 Discussion and Error Analysis

An example of the top candidates for a successful and an incorrect transliteration are given in Tables 4 and 5 respectively. We can see that the topic model has succeeded in finding the correct transliteration of ‘batik’, a traditional Javanese fabric, however a low score was given to the Korean transliteration of the name ‘Bernard’ appearing in the training data.

The benefits of the addition of a topic model is made clear with the example of ‘batik’ in Table 4. The semantic similarity measures give a higher score to ‘batik’ than ‘Batic’, a Slavic surname, despite ‘Batic’ being the more likely transliteration according to the baseline.

The improvement over the baseline for back-transliteration (XX-EN), on average +0.24, was considerably greater than that for transliteration (EN-XX), on average +0.17. We believe that this is due to the vast range of transliteration spelling variations in the non-English target languages. Since there is only one correct spelling variation defined in our test data and the topic distributions for each spelling variation are very similar, it is not possible to guess correctly. For an example of this problem, see Table 5.

### 7.1 Topic Alignment Difficulties

The majority of transliteration errors were caused by unsuccessful topic alignment between the source and target words. This was partly caused by the differences in usage of the original English words and the transliterated Japanese or Korean. For example, the



Candidate	Baseline	Cos	Cue	KL	SVM
<b>batik</b>	-1.29	<b>0.989</b>	<b>2.54e-04</b>	<b>-0.327</b>	<b>1.10</b>
baetic	-1.32	0.0764	1.67e-06	-1.65	-1.39
batic	<b>-0.708</b>	0.00	0.0	0.0	-1.48
batick	-0.788	0.00	0.0	0.0	-1.53
butic	-1.09	0.00	0.0	0.0	-1.68

Table 4: A good transliteration – バティック (*batikku* / ‘batik’) → batik.

Candidate	Baseline	Cos	Cue	KL	SVM
베르나르 <i>bereunareu</i>	-2.96	<b>0.642</b>	<b>4.78e-04</b>	<b>-1.72</b>	<b>0.112</b>
베르나르드 <i>bereunareudeu</i>	-3.65	0.243	3.84e-05	-2.41	-0.909
베른하르트 <i>bereunhareuteu</i>	-3.58	0.188	7.64e-05	-1.81	-0.969
<b>베르나르트 <i>bereunareuteu</i></b>	-4.24	0.217	8.24e-05	-2.69	-1.02
버나드 <i>beonadeu</i>	<b>-2.78</b>	0.123	4.33e-05	-3.01	-1.23

Table 5: An incorrect transliteration – bernard → 베르나르트 (*bereunareuteu*).

Japanese バイキング (*baikingu*) is a transliteration of ‘Viking’, however it is almost always used to mean ‘buffet’, deriving from the Scandinavian smorgasbord. In this case, we can expect the Japanese to be associated with food-related topics, quite different from ‘Viking’.

There are also many cases where words that do not clearly fit into one topic have unclear distributions across many groups. For example, the word 로마 (*roma* / ‘Rome’) could be more strongly categorized with ‘cities’ and ‘sightseeing’ in English but ‘history’ and ‘classical civilization’ in Korean, giving a low overall topic correlation.

## 7.2 Effect of Word Length and Frequency

We found that our model was more successful at finding the correct transliteration of longer words, as smaller words tend to have more spelling variations and are orthographically more similar to other words. By removing words of length 5 characters or less from the test data, we were able to improve the top-1 accuracy (SVM) to 0.593 (KO–EN, +0.089) and 0.721 (JA–EN, +0.111). In a practical lexicon extraction task over the entirety of Wikipedia this would cover roughly 35–45% of words (depending on language).

There was almost no variation in transliteration accuracy based on word frequency. The baseline is relatively unaffected by word frequency, with the exception of finding very rare character phrases not in the training data, and

the topic model proved to be robust across words of both high and low frequency.

## 8 Conclusion and Future Work

In this paper we demonstrated that the addition of semantic features can significantly improve transliteration accuracy. Specifically, it is possible to outperform the top-1 accuracy of a state-of-the-art phrase-based SMT transliteration baseline through the addition of a bilingual topic model.

Furthermore, our extended model is able to produce a considerable improvement in accuracy even for out-of-domain source words that have an unknown topic distribution. The experimental data set was constructed to simulate the task of extracting unknown word pairs from a comparable corpus, however our extension model results suggest that it will be possible to extract high-quality transliterations from larger and less comparable corpora than ever before.

In the future we would like to explore in depth the improvements to machine translation made possible by this approach.

## Acknowledgements

We would like to thank the anonymous reviewers for their feedback. The first author is supported by a Japanese Government (MEXT) research scholarship.

## References

- P.J. Antony, V.P. Ajith, and K.P. Soman. 2010. Statistical Method for English to Kannada Transliteration. *BAIP 2010, CCIS 70*, pp. 356–362.
- David Blei, Andrew Ng and Michael Jordan. 2003. Latent Dirichlet Allocation. In *The Journal of Machine Learning Research*, Volume 3.
- Eric Brill, Gary Kacmarcik and Chris Brockett. 2001. Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, 2001.
- Surya Ganesh, Sree Harsha, Prasad Pingali, Vasudeva Varma. 2008. Statistical Transliteration for Cross Language Information Retrieval using HMM alignment model and CRF. In *2nd International Workshop on Cross Language Information Access, IJCNLP 2008*.
- Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick and Dan Klein. 2008. Learning Bilingual Lexicons from Monolingual Corpora. In *ACL 2008*.
- Masato Hagiwara and Satoshi Sekine. 2011. Latent Class Transliteration based on Source Language Origin. In *ACL 2011*.
- Masato Hagiwara and Satoshi Sekine. 2012. Latent Semantic Transliteration using Dirichlet Mixture. In *ACL 2012*.
- Gumwon Hong, Min-Jeong Kim, Do-Gil Lee and Hae-Chang Rim. 2009. A Hybrid Approach to English-Korean Name Transliteration. In *Proceedings of 2009 Named Entities Workshop, ACL-IJCNLP*.
- Sittichai Jiampojarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim and Grzegorz Kondrak. 2010. Transliteration Generation and Mining with Limited Training Resources. In *Proceedings of the 2010 Named Entities Workshop, ACL 2010*.
- Nobuhiro Kaji and Masaru Kitsuregawa. 2011. Splitting Noun Compounds via Monolingual and Bilingual Paraphrasing: A Study on Japanese Katakana Words. In *EMNLP 2011*.
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Volume 1*.
- 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 *ACL 2007*.
- Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, Minghui Dong. 2007. Semantic Transliteration of Personal Names. In *ACL 2007*.
- David Matthews. 2007. Machine Transliteration of Proper Names. *Masters Thesis, School of Informatics, University of Edinburgh*.
- David Mimno, Hanna Wallach, Jason Naradowsky, David Smith and Andrew McCallum. 2009. Polylingual topic models. In *EMNLP 2009*.
- Toshiaki Nakazawa, Daisuke Kawahara and Sadao Kurohashi. 2005. Automatic Acquisition of Basic Katakana Lexicon from a Given Corpus. In *IJCNLP 2005*.
- Xiaochuan Ni, Jian-Tao Sun, Jian Hu, Zheng Chen. 2009. Mining Multilingual Topics from Wikipedia. In *WWW 2009*.
- Sara Noeman and Amgad Madkour. 2010. Language independent transliteration mining system using finite state automata framework. In *Proceedings of the 2010 Named Entities Workshop*.
- Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *ACL 2003*.
- Franz Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. In *Computational Linguistics 2003*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of Machine Translation. *Technical Report RC22176, IBM*.
- Xuan-Hieu Phan and Cam-Tu Nguyen. 2007. GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA).
- Reinhard Rapp. 1995. Identifying Word Translations in Non-Parallel Texts. In *ACL 1995*.
- Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of ICSLP, Volume 2*.
- Akihiro Tamura, Taro Watanabe and Eiichiro Sumita. 2012. Bilingual Lexicon Extraction from Comparable Corpora Using Label Propagation. In *EMNLP-CoNLL 2012*.
- Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
- Ivan Vulić, Wim De Smet and Marie-Francine Moens. 2011. Identifying Word Translations from Comparable Corpora Using Latent Topic Models. In *ACL 2011*.

# SUMT: A Framework of Summarization and MT

**Houda Bouamor**

**Behrang Mohit**

**Kemal Oflazer**

Carnegie Mellon University  
Doha, Qatar

hbouamor@qatar.cmu.edu, behrang@cmu.edu, ko@cs.cmu.edu

## Abstract

We present a novel system combination of machine translation and text summarization which provides high quality summary translations superior to the baseline translation of the entire document. We first use supervised learning and build a classifier that predicts if the translation of a sentence has high or low translation quality. This is a reference-free estimation of MT quality which helps us to distinguish the subset of sentences which have better translation quality. We pair this classifier with a state-of-the-art summarization system to build an MT-aware summarization system. To evaluate summarization quality, we build a test set by summarizing a bilingual corpus. We evaluate the performance of our system with respect to both MT and summarization quality and, demonstrate that we can balance between improving MT quality and maintaining a decent summarization quality.

## 1 Introduction

Machine Translation (MT) has been championed as an effective technology for knowledge transfer from English to languages with less digital content. An example of such efforts is the automatic translation of English Wikipedia to languages with smaller collections. However, MT quality is still far from ideal for many of the languages and text genres. While translating a document, there are many poorly translated sentences which can provide incorrect context and confuse the reader. Moreover, some of these sentences are not as *informative* and could be summarized to make a more cohesive document. Thus, for tasks in which complete translation is not mandatory, MT can be effective if the system can provide a

more informative subset of the content with higher translation quality.

In this work, we demonstrate a framework of MT and text summarization which replaces the baseline translation with a proper summary that has higher translation quality than the full translation. For this, we combine a state of the art English summarization system and a novel framework for prediction of MT quality without references.

Our research contributions are:

- (a) We extend a classification framework for reference-free prediction of translation quality at the sentence-level.
- (b) We incorporate MT knowledge into a summarization system which results in high quality translation summaries.
- (c) For evaluation purposes, we conduct a bilingual manual summarization of a parallel corpus.<sup>1</sup>

Our English-Arabic system reads in an English document along with its baseline Arabic translation and outputs, as a summary, a subset of the Arabic sentences based on their informativeness and also their translation quality. We demonstrate the utility of our system by evaluating it with respect to both its MT and the summarization quality. For summarization, we conduct both reference-based and reference-free evaluations and observe a performance in the range of the state of the art system. Moreover, the translation quality of the summaries shows an important improvement against the baseline translation of the entire documents.

This MT-aware summarization can be applied to translation of texts such as Wikipedia articles.

<sup>1</sup>The bilingually summarized corpora could be found at: <http://nlp.qatar.cmu.edu/resources/SuMT>

For such domain-rich articles, there is a large variation of translation quality across different sections. An intelligent reduction of the translation tasks results in improved final outcome. Finally, the framework is mostly language independent and can be customized for different target languages and domains.

## 2 Related work

Our approach draws on insights from problems related to text summarization and also automatic MT evaluation. Earlier works on Arabic summarization in campaigns and competitions such as DUC (Litkowski, 2004) or Multi-Ling (Gianakopoulos et al., 2011) were focused on abstractive summarization which involves the generation of new sentences from the original document. The fluency of such generated summaries might not be perfect. However, having a noisy source language text for an MT system can degrade the translation quality dramatically. Thus, extractive summarization like our framework is more suitable for MT summarization. In retrospect our annotated Arabic-English summaries is a unique bilingual resource as most other Arabic-English summarization corpora (e.g. DUC) are abstractive summaries.

There has been a body of recent work on the reference-free prediction of translation quality both as confidence estimation metrics and also direct prediction of human judgment scores (Bojar et al., 2013; Specia, 2012) or the range of the BLEU score (Soricut and Echiabi, 2010; Mohit and Hwa, 2007). These works mostly use supervised learning frameworks with a rich set of source and target language features. Our binary classification of MT quality is closer to the classification system of Mohit and Hwa (2007) to estimate translation difficulty of phrases. However, there are several modifications such as the method of labeling, the focus on sentence level prediction and finally the use of a different metric for both the labeling and final evaluation (which reduces the metric bias). For learning features, we cumulatively explore and optimize most of the reported features, and add document-level features to model the original document properties for each sentence.

Another line of research constrained by the lack of access to reference translations is confidence estimation for MT which is simply system's judg-

ments of its own performance. The confidence measure is a score for N-grams (substrings of the hypothesis) which are generated by an MT system. Confidence estimation is performed at the word level (Blatz et al., 2003) or phrase level (Zens and Ney, 2006). The measure is based on feature values extracted from the underlying SMT system and also its training data. There are many overlaps between the features used in confidence estimation and the MT quality prediction. However, the two frameworks use different learning methods. Confidence estimation systems usually do not have gold standard data and are mostly a linear interpolation of a large group of scores. In contrast, MT quality predictors such as our framework usually use supervised learning and rely on gold standard data.

Text summarization has been successfully paired with different NLP applications such as MT in cross-language summarization. Wan et al. (2010) and Boudin et al. (2011) proposed cross-language summarization frameworks in which for each sentence, in a source language text, an MT quality and informativeness scores are combined to produce summary in a target language (Chinese and French, respectively). In the latter, sentences are first translated, ranked and then summaries are generated. Differently, in Wan et al. (2010), each sentence of the source document is ranked based on an *a posteriori* combination of both scores. The selected summarized sentences are then translated to the target language using Google Translate. In contrast, we go a step further and design a hybrid approach in which we incorporate our MT quality classifier into the state-of-the-art summarization system. Moreover, we use SMT beyond a black-box and actually incorporate its knowledge in prediction of translation quality along with other set of features such as document-related and Arabic morphological information. Finally we demonstrate that our approach outperforms Wan et al. (2010) by conducting automatic evaluation of MT and summarization systems.

## 3 An overview of the approach

Given a source language document and its translation, our aim is to find a high quality summary of the translation with a quality superior to translating the entire document. Figure 1 illustrates an overview of our framework composed of the following major components: (a) a standard

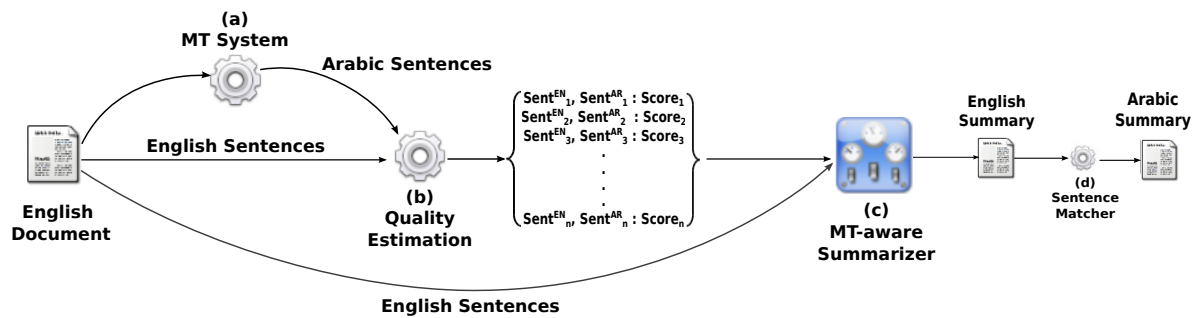


Figure 1: An overview of our MT-aware summarization system

SMT system; (b) our reference-free MT quality estimation system; (c) our MT-aware summarization system; and (d) the English-Arabic sentence matcher. Our system provides the translation summary through the following steps:

1. We translate an input English document into Arabic using the SMT system.
2. The quality estimation system (b) predicts if a translated sentence has high or low translation quality and assigns a quality score to each sentence.
3. We summarize the English document using our MT-aware summarization system (c), which incorporates the translation quality score (output of (b)) in its sentence selection process.
4. We produce the final Arabic translation summary by matching the English summarized sentences with the corresponding Arabic translations (d).
5. We automatically evaluate the quality of our MT-aware summarization system using MT and summarization metrics.

Our contributions are mainly related to the second and third components which will be discussed in Sections 4 and 5.

#### 4 Reference-free quality estimation of MT

Our system needs to estimate the translation quality without access to the Arabic reference translations. The reference-free MT evaluation has been investigated extensively in the past decade. A valuable gold-standard resource for many of these studies are human judgment scores which have

been developed in evaluation programs like NIST, and workshops such as WMT (Koehn and Monz, 2006). Since such human judgments do not exist for English to Arabic translations, we adapt the framework of Mohit and Hwa (2007) for predicting the translation quality. This framework uses only reference translations and the automatic MT evaluation scores to create labeled data for training a classifier. The binary classifier reads in a source language sentence, with its automatically obtained translation and predicts if the target sentence has *high* or *low* translation quality. We describe details of this framework in the following section.

##### 4.1 Labeling gold-standard data

In order to train the binary classifier, we need gold standard data with English source sentences labeled as having high or low translation quality when translated into Arabic. For this labeling, we estimate translation quality by the Translation Edit Rate TER metric (Snover et al., 2006).<sup>2</sup> We deliberately use two different metrics for gold standard labeling (TER) and the final MT evaluation (using BLEU (Papineni et al., 2002)) to reduce the bias that a metric can introduce to the framework. In this task, we use a parallel corpus that is composed of a set of documents. We automatically translate each document and label its sentences based on the following procedure:

- (a) Measure the TER score of the document against its reference translation.
- (b) For each sentence within the document, measure its TER score: If this score is higher than the document score, it has *low* translation quality. Otherwise it has *high* translation quality.

<sup>2</sup>This automatic labeling framework exempts us from the manual labeling of translation quality like Wan et al. (2010).

This provides a simple estimate of the translation quality for a source language sentence relative to the document that it belongs to. This document-level relevance is a deliberate choice to build a classifier that ranks the translation quality of a sentence with respect to other sentences in the document (similar to a summarization system). We also note that the quality labeling is obviously non-absolute and relative to the specific SMT engine used in this work.

## 4.2 MT quality classifier

We use a Support Vector Machine (SVM) classifier and exploit a rich set of features to represent a source language sentence and its translation. We use the default configuration with a linear kernel function. In order to estimate a score for the translation quality, we use a normalized form of the classifier's score for each sentence. The score is the distance from the separating surface and is proper estimate of the intensity of the class label.

## 4.3 Learning features

We use a suite of features that have been extensively used in works related to translation quality estimation. We adapt the feature extraction procedure from the Quest framework (Specia et al., 2013) to our English-Arabic translation setup, and extract the following groups of features:

**General features:** For each sentence we use different features modeling its length in terms of words, the ratio of source-target length, source-target punctuation marks, numerical characters, and source-target content words.

**Language model scores:** The likelihood of a target language (Arabic) sentence can be a good indicator of its grammaticality. In our experiments, we used the SRILM toolkit (Stolcke, 2002) to build 5-gram language model using the LDC Arabic Gigaword corpus. We then, apply this model to obtain log-likelihood and perplexity scores for each sentence.

**MT-based scores:** We extract a set of features from the generated MT output. These include the absolute number and the ratios of out of vocabulary terms and the ratio of Arabic detokenization that is performed on the Arabic MT output.

**Morphosyntactic features:** We use features to model the difference of sequences of POS tags

for a pair of source-target sentences. These features measure the POS preservation in the translation process (e.g. measuring if the proper nouns in the source sentence are kept and also translated as proper nouns in Arabic). We compute the absolute difference between the number of different POS tags. The source and target sentences are tagged respectively using the TreeTagger (Schmid, 1994) and AMIRA (Diab, 2009) toolkits. We also, indicate the percentage of nouns, verbs, proper nouns in the source and target sentences.

**Document-level features:** We extend Mohit and Hwa (2007) framework by incorporating a set of document-level features (in addition to the sentence-level ones) which scales the sentence's classification relative to its document. In a linear model, these document-level features rescale and shift the feature space relative to the given document which helps us to classify the sentence with respect to the document. These features consist of the average of the sentence-level features described above.

## 5 MT-aware Summarization

We pair summarization and MT (SUMT) by including information about the MT quality into the summarization system. Our MT-aware summarizer focuses on the linguistic and translation quality of a given sentence, as well as its position, length, and the content in its sentence ranking procedure. The main goal of this system is to obtain an informative summary of a source document with an improved translation quality that could replace the complete, yet less fluent translation of the document.

We explore various configurations and find the sweet spot of the translation and summarization qualities in the system illustrated in Figure 1. This includes converting the MEAD summarizer into an MT-aware summarization framework by including information from the classifier into the sentence ranking procedure.

### 5.1 The MEAD Summarization system

In our experiments, the summary for each document is generated using MEAD (Radev et al., 2004), a state-of-the-art single- and multi-document summarization system. MEAD has been widely used both as a platform for developing summarization systems and as a baseline system for testing novel summarizers. It is a

centroid-based extractive summarizer which selects the most important sentences from a sequence of sentences based on a linear combination of three parameters: the sentence length, the centroid score and the position score (Radev et al., 2001). MEAD also employs a cosine reranker to eliminate redundant sentences. We create summaries at 50% length (a fixed ratio for all documents) using MEAD’s default configurations.

## 5.2 SUMT system

Our MT-aware summarizer (SUMT) represents an approach of adapting the basic sentence scoring/ranking approach of MEAD. We extend the default MEAD sentence ranking procedure by incorporating information about the translation quality of the sentence. This score is provided by our SVM-based classifier. The selected sentences generally correspond to those having high translation quality (estimated by TER).

Typically, the ranking score of a sentence is defined by a linear combination of the weighted sentence position, centroid and length scores. We used the default weights defined for each feature in the default version of MEAD. The additional quality feature weight is optimized automatically towards the improvement of BLEU, using a held-out development set of documents. Finally, sentences in each document are ranked based on the final obtained score. In this work, we take a hard 50% summarization ratio which is applied to MEAD, SUMT and our gold standard summaries.

## 6 Experimental Setup

In this section, we explain details of the data and the general setting for different components of our system.

### 6.1 Translation and Summarization Corpora

For our experiments, we use the standard English-Arabic NIST test corpora which are commonly used MT evaluations.<sup>3</sup> We use the documents provided in NIST 2008 and 2009 for the training and development, and those in the NIST 2005 for testing. Each collection contains an Arabic and four English reference translations. Since we work on English to Arabic translation, we only use the first translation as the reference.

<sup>3</sup>All of the different MT corpora can be accessed from Linguistic Data Consortium (LDC).

### 6.2 Annotation of gold-standard summaries

The automatic summarization should be able to reduce the complexity of documents length wise, while keeping the essential information from the original documents like important events, person names, location, organizations and dates. In order to evaluate the quality of the summaries, we conducted a bilingual summarization of our test corpus (the NIST 2005). This parallel corpus is composed of 100 parallel documents containing each in average 10 sentences. We asked two native speakers (one per language) to summarize each side of the corpora independent of each other and independent of the MT output. We set a hard 50% ratio for annotators to choose approximately half of the sentences per document. Annotators followed a brief guideline to completely understand the entire document and examine and select summary sentences based on the following criteria: (a) Being informative with respect to the main story and the topic (b) minimizing the redundancy of information (c) preserving key information such as the named entities and dates. We obtain as inter-judge agreement a value of  $\kappa = 0.61$  corresponding to a moderate agreement according to the literature.<sup>4</sup>

### 6.3 MT Setup

The baseline MT system is the open-source MOSES phrase-based decoder trained on a standard English-Arabic parallel corpus. This 18 million word parallel corpus consists of the non-UN parts of the NIST corpus distributed by the LDC. We perform the standard preprocessing and tokenization on the English side using simple punctuation-based rules. We also use the MADA+TOKAN morphological analyzer (Habash et al., 2009) to preprocess and tokenize the Arabic side of the corpus. The corpus is word-aligned using the standard setting of GIZA++ and the grow-diagonal-final heuristic of MOSES. We use the 5-gram language model with modified Kneser-Ney smoothing. The language model for our system is trained using the LDC Arabic Gigaword corpus. A set of 500 sentences is used to tune the decoder parameters using the MERT (Och, 2003). After decoding, we use the El Kholy and Habash (2010) Arabic deto-

<sup>4</sup>This Cohen’s kappa value is obtained using the MEAD evaluation tool designed to assess the agreement between two summaries.

kenization framework to prepare the Arabic output for evaluation.

#### 6.4 MT-quality classifier

We use the models described in Section 4 to build a Support Vector Machine (SVM) binary classifier using the LIBSVM package (Chang and Lin, 2011). To train our classifier we use a total of 2670 sentence pairs extracted from 259 documents of NIST 2008 and 2009 data sets. The sentences are labeled following our TER-based procedure. The automatic labeling procedure (section 4.1) enforces a rough 50-50 high and low quality translations. Thus, we obtained 1370 negative examples and 1363 positive ones. For all tests, we use a set of 100 documents from the NIST 2005 test set, containing 1056 sentences.

### 7 Evaluation and results

We experimented with different configurations of the MT and the summarization system with the goal of achieving a balanced performance in both dimensions. We reached the sweet spot of performance in both dimensions in our MT-aware summarization system in which we achieved major (over 4 points BLEU score) improvements while maintaining an acceptable summarization quality. In the following we discuss the performance of the MT and summarization systems.

#### 7.1 MT evaluation

Table 1 presents MT quality for the baseline system and different summarization frameworks measured by BLEU, TER and METEOR (Lavie and Agarwal, 2007) scores.<sup>5</sup>

The remaining MT experiments are conducted on summarized documents. These include summaries provided by: (a) a length-based baseline system that simply chooses the subset of sentences with the shortest length (**Length**); (b) the state of the art MEAD summarizer (**MEAD**); (c) our MT quality estimation classifier (**Classifier**); (d) a linear interpolation of informativeness and MT quality scores in the spirit of Wan et al. (2010) (**Interpol**)<sup>6</sup>; (e) our MT-aware summarizer

<sup>5</sup>Our English to Arabic baseline system shows a performance in the ballpark of the reported score for the state of the art systems (e.g. El Kholly and Habash (2010)).

<sup>6</sup>The overall score of a sentence is defined as follows:  $score = (1 - \lambda) * InfoScore + \lambda * TransScore$  where  $\lambda = 0.3$  and  $TransScore$  and  $InfoScore$  denote the MT quality score and the informativeness score of a sentence.

(**SuMT**); and (f) an oracle classifier which chooses the subset of sentences with the highest translation quality (**Oracle**). This oracle provides an upper bound estimate of room that we have to improve translation quality of the summaries.

	BLEU	TER	METEOR
<i>Baseline</i>	27.52	58.00	28.51
<b>Length</b>	26.33	58.13	27.81
<b>MEAD</b>	28.42	55.00	28.82
<b>Classifier</b>	31.36	52.00	29.22
<b>Interpol</b>	28.45	55.00	29.05
<b>SuMT</b>	<b>32.12</b>	<b>51.00</b>	<b>30.48</b>
<b>Oracle</b>	34.75	47.00	32.42

Table 1: A comparison of MT quality for full and summarized documents.

We are aware that the comparison of the MT baseline system with these summarization systems is not a completely fair comparison as the test sets are not comparable. However, with a ballpark comparison of the baseline (for full documents) with the summarized documents, we demonstrate the average range of improvement in translation quality. Moreover, we compare different summarization systems with each other to reach the best combination of MT and summarization quality.

We set a 50% summarization ratio in all experiments and also in creation of the gold-standard to create similar comparable conditions. For example, for evaluating our quality estimation classifier as a summarizer, we filter out the bottom 50% of the sentences (based on their classification scores) for each document and evaluate the translation quality of the top 50% Arabic translation sentences.

The MT results for the MEAD summarizer indicate that summarization of MT does not necessarily improve MT quality. In contrast, the comparison between the baseline, the oracle summarizer and SuMT system demonstrates a major improvement in MT quality that is competitive with the oracle summarizer (an improvement of almost +5 BLEU scores). The results given in Table 1 show also that our system produce better MT quality sentences than Interpol (+4.67 BLEU points). This could be explained by the higher weight assigned to the informativeness score in the linear interpolation. In the following sections we demonstrate that we maintain a decent summarization quality while we achieve these MT improvements.



	<i>English</i>					<i>Arabic</i>				
	<b>Length</b>	<b>MEAD</b>	<b>Classifier</b>	<b>Interpol</b>	<b>SuMT</b>	<b>Length</b>	<b>MEAD</b>	<b>Classifier</b>	<b>Interpol</b>	<b>SuMT</b>
<b>ROUGE-1</b>	54.21	<b>75.93</b>	67.41	73.72	72.51	36.01	45.66	44.94	45.33	<b>46.43</b>
<b>ROUGE-2</b>	38.15	<b>67.77</b>	56.72	66.01	62.83	15.19	22.83	22.23	22.46	<b>23.28</b>
<b>ROUGE-SU4</b>	38.99	<b>67.96</b>	57.03	54.14	63.17	15.81	23.56	23.09	20.33	<b>24.07</b>
<b>ROUGE-L</b>	51.77	<b>74.92</b>	65.92	72.79	71.17	33.74	43.20	42.33	42.81	<b>43.84</b>

Table 2: ROUGE F-Scores for different summarization systems providing 50% length for English and Arabic summaries for each document.

## 7.2 Model-based summarization evaluation

We evaluate the quality of our summarization systems for both English and Arabic. We first focus on English summaries generated using different summarization configurations, and then evaluate the quality of Arabic summaries obtained by matching the English summarized sentences with the corresponding Arabic translations. It is not surprising that summarizing a noisy Arabic MT output would not produce high quality Arabic summaries. Instead, we use the parallel corpus to project the summarization from the source language (English) to the corresponding Arabic translations.

For evaluating our summarization systems, we use ROUGE (Lin, 2004), a metric based on n-gram similarity scores between a model summary generated by human and an automatically generated peer summary. We use the ROUGE-1, ROUGE-2, ROUGE-SU4 and ROUGE-L F-scores with the two human summaries described in Section 6 as models.<sup>7</sup> We use the same parameters and options in ROUGE as in the DUC 2007 summarization evaluation task.<sup>8</sup> Table 2 presents the ROUGE F-scores obtained on our test datasets for the different summarization systems for both languages.

Similar to section 7.1, we experiment with five summarizers: **Length**, **MEAD**, **Classifier**, **Interpol**, **SuMT**. As expected, the MEAD summarizer shows the best summarization performance. Also, the length-based baseline system generates poor quality summaries (about 22 score ROUGE-1 reduction from MEAD). This is not surprising since the baseline only uses the length of the sentence regardless its content. Furthermore, the perfor-

<sup>7</sup>A study conducted by Lin and Hovy (2003) shows that automatic evaluation using unigram and bigram co-occurrences between summary pairs have the highest correlation with human evaluations and have high recall and precision in significance test with manual evaluation results.

<sup>8</sup><http://duc.nist.gov/duc2007/tasks.html>.

mance of the classifier-based summarizer is lower than the MEAD, because it does not use the summarization feature and only relies on an estimated translation quality to select the sentences.

Reviewing different values of the ROUGE metric in the left side Table 2, we observe that SuMT and Interpol summaries maintain a decent quality, comparable to the state of the art MEAD. For example, they give promising results in terms of ROUGE-L (71.17% and 72.79%, respectively), which consistently indicates that the sentences produced are closer to the reference summary in linguistic surface structure than those of the classifier (65.92). In addition to the quality of the English summaries, we are more interested in assessing the quality of the Arabic summaries. This comes back to our main goal of producing a fluent Arabic summary with good translation quality. We evaluate the Arabic summaries by measuring different ROUGE metrics against our model summaries. The results in the right side of Table 2 show that our MT-aware summarization framework achieves the best results in different ROUGE configurations and outperforms the state-of-the-art summarizer (+1 point ROUGE-1). In other words, our Arabic translated summaries generated using SuMT, are the most fluent and have the most similar structure compared with the Arabic model summaries.

## 7.3 Model-free summarization evaluation

In addition to the reference-based summarization evaluation described above, we conducted model-free experiments evaluating the summary quality for both languages. Recently, Louis and Nenkova (2013) proposed SIMetrix, a framework that does not require gold standard summaries for measuring the summarization quality. The framework is based on the idea that higher similarity with the source document would be indicative of high quality summary. SIMetrix is a suite of model-free similarity metrics for comparing a generated sum-

mary with the source document for which it was produced. That includes cosine similarity, distributional similarity and also use of topic signature words. SIMetrix is shown to produce summary scores that correlate accurately with human assessments.<sup>9</sup>

We used SIMetrix to evaluate the quality of the summaries generated by different systems. We report in Table 3, **%TopicTokens** referring to the percentage of tokens in the summary that are topic words of the input document; the Kullback Leibler divergence (**KL**); and the Jensen Shannon divergence (**JS**) between vocabulary distributions of the input and summary texts, which was found to produce the best predictions of summary quality. Since KL divergence is not symmetric, we measure it both ways Input-Summary (**KL<sub>IS</sub>**) and Summary-Input (**KL<sub>SI</sub>**). Based on these metrics, a good summary is expected to have low divergence between probability distributions of words in the input and summary, and high similarity with the input.

Table 3 illustrates these similarity results for both English and Arabic summaries. The results are consistent with those found in the model-based evaluation. For Arabic, our MT-aware system achieves the best results in terms of different divergence (0.14 JS against 0.17 for MEAD) and topic related scores (73.37% of tokens in the SUMT Arabic summaries are topic words in the input document against 71.78% in MEAD summaries). It is important to note that lower divergence scores indicate higher quality summaries.

	Length	MEAD	Classifier	Interpol	SuMT
<i>English</i>					
<b>%TopicTokens</b>	63.21	63.70	63.28	63.50	63.33
<b>KL<sub>IS</sub></b>	0.37	0.18	0.33	0.19	0.25
<b>KL<sub>SI</sub></b>	0.14	0.02	0.12	0.07	0.09
<b>JS</b>	0.04	0.01	0.03	0.02	0.02
<i>Arabic</i>					
<b>%TopicTokens</b>	71.51	71.61	73.18	72.44	<b>73.37</b>
<b>KL<sub>IS</sub></b>	1.30	1.24	1.28	1.20	<b>1.19</b>
<b>KL<sub>SI</sub></b>	1.11	1.07	1.03	0.96	<b>0.94</b>
<b>JS</b>	0.17	0.15	0.16	0.15	<b>0.14</b>

Table 3: Distribution similarity scores for each system summaries evaluated against the input document for English and Arabic.

<sup>9</sup>SIMetrix is available at: <http://www.seas.upenn.edu/~lannie/IEval2.html>.

## 8 Conclusion and Future work

We presented our approach in pairing automatic text summarization with machine translation to generate a higher quality content. We demonstrated an English to Arabic MT aware summarization framework with high summarization quality and greatly improved translation quality.

We plan to extend our current system in the following directions: (a) We will examine alternative learning frameworks and features to improve our prediction of the translation quality. (b) We will explore different methods to incorporate and optimize the MT quality information with the summarization system. (c) We will explore alternative text domains such as Wikipedia in which there is a larger variation of translation quality in different parts of the document. Considering the poor translation quality of many language pairs, text summarization can provide effective support for MT in various end-user applications. We believe there are many avenues to explore in this direction of research.

## 9 Acknowledgements

We thank Nizar Habash and anonymous reviewers for their valuable comments and suggestions. We thank Mollie Kauffer and Wajdi Zaghouni for preparing the Arabic and English summaries. This publication was made possible by grants YSREP-1-018-1-004 and NPRP-09-1140-1-177 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## References

- John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence Estimation for Machine Translation.
- Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the ACL-WMT-2013*.
- Florian Boudin, Stéphane Huet, and Juan-Manuel Torres-Moreno. 2011. A Graph-based Approach to Cross-language Multi-document Summarization. *Polibits*, (43):113–118.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIB-SVM: A Library for Support Vector Machines.

- ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.
- Mona Diab. 2009. Second generation AMIRA Tools for Arabic Processing: Fast and Robust Tokenization, POS Tagging, and Base Phrase Chunking. In *Proceedings of MEDAR*, Cairo, Egypt.
- Ahmed El Kholy and Nizar Habash. 2010. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proceedings of LREC*.
- George Giannakopoulos, Mahmoud El-Haj, Benoît Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. 2011. TAC 2011 Multiling Pilot Overview. In *Proceedings of the TAC 2011*.
- Nizar Habash, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, Pos Tagging, Stemming and Lemmatization. In *Proceedings of MEDAR*.
- Philipp Koehn and Christof Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings of the Workshop on Statistical Machine Translation*.
- 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 WMT*.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of NAACL*.
- Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the ACL-04 Text Summarization Workshop (Text Summarization Branches Out)*.
- Kenneth C Litkowski. 2004. Summarization Experiments in DUC 2004. In *Proceedings of the HLT-NAACL Workshop on Automatic Summarization, DUC-2004*, pages 6–7.
- Annie Louis and Ani Nenkova. 2013. Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, 39(2):1–34.
- Behrang Mohit and Rebecca Hwa. 2007. Localization of Difficult-to-Translate Phrases. In *Proceedings of ACL WMT-07*.
- Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*.
- Dragomir Radev, Sasha Blair-Goldensohn, and Zhu Zhang. 2001. Experiments in single and multi-document summarization using MEAD. In *Proceedings of DUC*.
- Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. 2004. MEAD-a platform for multidocument multilingual text summarization. In *Proceedings of LREC*.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*.
- Matthew Snover, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA*.
- Radu Soricut and Abdessamad Echihabi. 2010. TrustRank: Inducing Trust in Automatic Translations via Ranking. In *Proceedings of ACL*.
- Lucia Specia, Kashif Shah, Jose Guilherme Cargomo de Souza, and Trevor Cohn. 2013. QUEST-A Translation Quality Estimation Framework. In *Proceedings of the ACL, demo session*, Sofia, Bulgaria.
- Lucia Specia. 2012. Estimating Machine Translation Quality. In *MT Marathon*.
- Andreas Stolcke. 2002. SRILM-an Extensible Language Modeling Toolkit. In *Proceedings of ICLSP*.
- Xiaojun Wan, Huiying Li, and Jianguo Xiao. 2010. Cross-language Document Summarization Based on Machine Translation Quality Prediction. In *Proceedings of ACL*, Uppsala, Sweden.
- Richard Zens and Hermann Ney. 2006. N-Gram Posterior Probabilities for Statistical Machine Translation. In *Proceedings of WMT*.

# Tuning SMT with A Large Number of Features via Online Feature Grouping

Lemao Liu<sup>1</sup>, Tiejun Zhao<sup>1</sup>, Taro Watanabe<sup>2</sup>, Eiichiro Sumita<sup>2</sup>

<sup>1</sup>School of Computer Science and Technology  
Harbin Institute of Technology, Harbin, China

<sup>2</sup>National Institute of Information and Communications Technology

3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, Japan

{lmliu|tjzhao}@mtlab.hit.edu.cn

{taro.watanabe|eiichiro.sumita}@nict.go.jp

## Abstract

In this paper, we consider the tuning of statistical machine translation (SMT) models employing a large number of features. We argue that existing tuning methods for these models suffer serious sparsity problems, in which features appearing in the tuning data may not appear in the testing data and thus those features may be over tuned in the tuning data. As a result, we face an over-fitting problem, which limits the generalization abilities of the learned models. Based on our analysis, we propose a novel method based on feature grouping via OSCAR to overcome these pitfalls. Our feature grouping is implemented within an online learning framework and thus it is efficient for a large scale (both for features and examples) of learning in our scenario. Experiment results on IWSLT translation tasks show that the proposed method significantly outperforms the state of the art tuning methods.

## 1 Introduction

Since the introduction of log-linear based SMT (Och and Ney, 2002), tuning has been a hot topic. Various methods have been explored: their objectives are either error rates (Och, 2003), hinge loss (Watanabe et al., 2007; Chiang et al., 2008) or ranking loss (Hopkins and May, 2011), and they are either batch training or online training methods. In this paper, we consider tuning translation models with a large number of features such as lexical, n-gram level and rule level features, where the number of features is largely greater than the number of bilingual sentences. Practically, existing tuning methods such as PRO and MIRA might

This joint work was done while the first author visited NICT.

be applied in our scenario, however, they will suffer from some pitfalls as well, which have been less investigated in previous works.

One of pitfalls is that these features are so sparse that many features which are potentially useful for a test set may not be included in a given tuning set, and many useless features for testing will be over tuned on the development set meanwhile. As a result, the generalization abilities of features are limited due to the mismatch between the testing data and the tuning data, and over-fitting occurs. One practice is to tune translation models on a larger tuning set, such as the entire training data (Xiao et al., 2011; Simianer et al., 2012), in the hope that more features would be included during tuning. However, tuning robust weights for translation models has additional requirements to a tuning set. Firstly, multiple reference translations in the tuning data are helpful for better tuning, especially when testing data contains multiple reference translations. Secondly, the closeness between the tuning set and a test set is also important for better testing performance (Li et al., 2010). These requirements can explain why tuning on the training data leads to unsatisfactory performance on the IWSLT translation task, as will be shown in our experiments later. Therefore, enlarging a tuning set is not always a sufficient solution for robust tuning, since it would be impractical to create a large scale tuning set with these requirements.

We propose a novel tuning method by *grouping* a large number of features to leverage the above pitfalls. Instead of directly taking the large number of atomic features into translation model, we firstly learn their group structure on the training data to alleviate their serious sparsity. Then, we tune the translation model consisting of grouped features on a multi-reference development set to ensure robust tuning. Unlike unsupervised clustering methods such as k-means (MacQueen, 1967) for feature clustering, we group the features with

the OSCAR (Octagonal Shrinkage and Clustering Algorithm for Regression) method (Bondell and Reich, 2008), which directly relates the objective of feature grouping to translation evaluation metrics such as BLEU (Papineni et al., 2002) and thus grouped features are optimized with respect to BLEU. Due to the large number of features and large number of training examples, efficient grouping is not simple. We apply the online gradient projection method under the FOBOS (forward-backward splitting) framework (Duchi and Singer, 2009) to accelerate feature grouping.

We employ a large number of features by treating each translation rule in a synchronous-CFG as a single feature. Experiments on IWSLT Chinese-to-English translation tasks show that, with the help of grouping these features, our method can overcome the above pitfalls and thus achieves significant improvements.

## 2 Tuning Method

We propose a novel tuning method for translation models with a large number of features, which incorporates feature grouping. Our assumption is that although a feature which is useful for a test set does not appear in the tuning set, another similar feature may exist. Therefore, grouping similar features can alleviate sparsity in this way. The proposed tuning method consists of two steps: first, it tries to learn a group structure for atomic features; second, it treats each feature group as a single feature and tunes the translation model on a given tuning set using off-the-shelf toolkits such as PRO. In the first step, we learn a group structure of atomic features in the large training data for better coverage. In the second step, we tune a translation model with the grouped features on a given development set with multiple references to ensure the robust tuning.

Before describing our tuning algorithm, we present notations for the rest of this paper. Suppose  $\mathcal{H}$  is a feature set consisting of atomic features  $\{h_1, h_2, \dots, h_d\}$  or their index set  $\{1, 2, \dots, d\}$  for simplicity;  $H = \langle h_1, h_2, \dots, h_d \rangle$  is a  $d$ -dimensional feature vector function with respect to  $\mathcal{H}$ , and  $W$  is its weight vector with each component  $W_i$  and dimension  $d$ ;  $\mathcal{G} = \{g_1, g_2, \dots, g_M\}$  is a group of  $\mathcal{H}$ , where each element  $g_i$  is a power set of  $\mathcal{H}$ . Similarly,  $G = \langle g_1, g_2, \dots, g_M \rangle$  is an  $M$ -dimensional feature vector function with respect to  $\mathcal{G}$  and  $W^{\mathcal{G}}$  is

its weight with each component  $W_i^{\mathcal{G}}$ . In this paper, we consider the disjoint  $\mathcal{G}$ , i.e.  $g_i \cap g_j = \emptyset$  if  $i \neq j$ . Further, suppose  $\Delta(W)$  is a set of the index  $i$  such that  $W_i \neq 0$ , and  $|\cdot|$  is either the number of elements in a set  $S$  or the absolute value of a real number  $x$ .

---

### Algorithm 1 Tuning Algorithm

---

**Input:** training data, dev,  $W^{ini}$ ,  $T$

- 1: Initialize  $W^1 = W^{ini}$
- 2: **for all**  $i$  such that  $1 \leq i \leq T$  **do**
- 3:   Decode on training data with  $W^i$  to obtain a k-best-list and merge k-best-lists
- 4:   Update the group set  $\mathcal{G}$  based on the merged k-best-list  $\triangleright$  **Call Algorithm 2**
- 5:   Tune the translation model with  $\mathcal{G}$  as the feature set on dev with PRO to update  $W^{\mathcal{G}}$
- 6:   Unpack  $W^{\mathcal{G}}$  to  $W^{i+1}$
- 7: **end for**
- 8:  $W = W^{T+1}$

**Output:**  $W$

---

Algorithm 1 describes our two-step tuning procedure for a translation model with  $\mathcal{H}$  as its feature set. It inputs a training data set, a development set, initial weight  $W^1$  with respect to  $\mathcal{H}$ , and maximal iterations  $T$ ; and outputs a weight  $W$ . It initializes with  $W^1$  in line 1; from line 2 to line 7, it iteratively obtains a k-best-list by decoding with  $W^i$ , updates the group set  $\mathcal{G}$ , tunes the translation weight  $W^{\mathcal{G}}$  based on  $\mathcal{G}$ , and unpacks the  $W^{\mathcal{G}}$  to obtain  $W^{i+1}$ . At the end, it returns the final weight  $W$ . In particular, the k-best-list is obtained using  $H$  as a feature vector with its weight vector  $W$  derived from grouped weights  $W^{\mathcal{G}}$  through unpacking: if  $h_j \in g_k$ , then  $W_j = W_k^{\mathcal{G}}$ . The grouping algorithm in line 3 will be introduced in the next section.

In this paper, we use a hierarchical phrase based translation model, which consists of 8 default features: translation probabilities, lexical translation probabilities, word penalty, glue rule penalty, synchronous rule penalty and language model. In addition, we also employ a large number rule identify (id) features: each rule itself is a feature, and if a translation contains a rule for  $x$  times, then the value of this rule id feature is  $x$ . In line 4 we group these id features and impose that each default feature itself is a group.

## 3 Online Feature grouping

---

**Algorithm 2 Feature Grouping Algorithm**

---

**Input:**  $\lambda_1, \lambda_2, k$ -best-list,  $W^1, n$ 

- 1: Collect a set of tuples  $\{\langle f, e', e^* \rangle\}$  from  $k$ -best-list
- 2: **for all**  $i$  such that  $1 \leq t \leq n$  **do**
- 3: Randomly select  $\langle f, e', e^* \rangle$  from the tuple set
- 4:  $W^{t+1/2} = W^t + \nabla_W \delta(f, e', e^*, W^t)/t$
- 5: Minimize  $Q(W; W^{t+1/2}, t+1, \lambda_1, \lambda_2)$  to obtain  $(W^{t+1}, \mathcal{G})$

▷ **Group optimization**

- 6: **end for**

**Output:**  $\mathcal{G}$ 

---

Suppose  $f$  is a sentence in a development set,  $C$  is a set of translations for  $f$ , and  $r$  is a set of reference translations for  $f$ . Following PRO, we define ranking loss function as follows:

$$L(W) = \frac{1}{N} \sum_f \sum_{e^*, e'} \delta(f, e', e^*, W), \quad (1)$$

with

$$\delta(f, e', e^*, W) = \max_W \left\{ (H(f, e') - H(f, e^*)) \cdot W + 1, 0 \right\},$$

where  $e', e^* \in C$  such that  $\text{BLEU}(e^*, r) > \text{BLEU}(e', r)$ , and  $N$  is the number of all tuples  $\langle e^*, e', f \rangle$ .

To achieve group structure and avoid the sparsity in  $H$ , we apply the OSCAR over the above loss function, and obtain the function:

$$L(W) + \lambda_1 \sum_{i=1}^d |W_i| + \lambda_2 \sum_{1 \leq i < j \leq d} \max\{|W_i|, |W_j|\}, \quad (2)$$

where  $d$  is the dimension of feature vector  $H$  or its weight  $W$ ,  $\lambda_1$  and  $\lambda_2$  are two hyperparameters for two regularizers taking positive value. Minimization of Eq.2 makes some components in  $W$  equal and thus achieves a feature grouping effect. In other words,  $W_i = W_j$  means that  $h_i$  and  $h_j$  lie in the same group, i.e.  $h_i, h_j \in g_k$  for some  $g_k \in \mathcal{D}(W)$ , where  $\mathcal{D}(W)$  denotes the group derived from  $W$  as follows. Given  $W$ , we first sort its components  $W_i$  to obtain a permutation  $\{i_k\}_{k=1}^d$  such that  $W_{i_1} \leq W_{i_2} \leq \dots \leq W_{i_d}$  with  $1 \leq i_k \leq d$ ; then we can easily obtain  $\mathcal{D}(W)$  after traversing  $\{W_{i_k}\}_{k=1}^d$ . For example,  $W =$

$\langle 1, 3, 1, 3, 1 \rangle$ , then  $\mathcal{D}(W) = \{\{1, 3, 5\}, \{2, 4\}\}$ . One advantage of OSCAR over unsupervised clustering methods (e.g.  $k$ -means) is that it relates the objective of grouping to an error metric, such as BLEU, and thus can achieve an optimal grouping towards BLEU.

Bondell and Reich (2008) firstly proposed two approaches for OSCAR. The first one casts the problem into a quadratic program (QP) consisting of  $O(d^2)$  variables and  $O(d^2)$  constraints. The second one tries to optimize a sequence of (potentially smaller) QP's with more constraints, which can be up to  $O(d!)$  in the worst case. Zhong and Kwok (2011) explored a much faster approach which is based on the accelerated gradient and projection method. Its complexity is reduced to  $O(d \log d)$ . Since the dimension  $d$  of  $H$  is large enough in our scenario where  $d$  is up to hundred of thousands, these existing optimization methods are inefficient to minimize Eq.2. Here, based on (Zhong and Kwok, 2011), we employ an online gradient projection algorithm under the FOBOS framework for faster learning. The framework of FOBOS is a type of online learning, in which it is theoretically guaranteed to solve such a problem as in Equation 2: the objectives consisting of two additive terms, in which one is non-smooth but convex and the other is smooth and convex<sup>1</sup>. FOBOS contains two steps: it first performs a gradient descent operator, and then updates weight by a proximity (or projection) operator.

Algorithm 2 describes the online training of feature grouping. It requires some inputs: two regularizer parameters  $\lambda_1$  and  $\lambda_2$ ; a  $k$ -best-list translations; an initial weight  $W^1$ ; and a maximum iterations  $n$ . It firstly collects a set of tuples encoded with translation pairs from  $k$ -best-list following the strategy implemented in the PRO toolkit in line 1. It repeatedly updates weight  $W^t$  and feature group  $\mathcal{G}$  from line 2 to line 6: it randomly samples a tuple  $\langle f, e', e^* \rangle$  from the collected tuple set in line 3, it performs a gradient descent operator in line 4 where  $\nabla_W \delta(f, e', e^*, W^t)$  denotes the subgradient of  $\delta(f, e', e^*, W^t)$  at current weight  $W^t$ , and it optimizes  $(W^{t+1}, \mathcal{G})$  by a proximity operator for group optimization in line 5. At last it returns the group result  $\mathcal{G}$ .

In particular, the subgradient of  $\delta(f, e', e^*, W^t)$

---

<sup>1</sup>For Eq.2, the non-smooth but convex term is the entire of Eq.2, and the smooth and convex term can be considered as 0.

in line 4 is defined via the following equation:

$$\nabla_W \delta(f, e', e^*, W) = \begin{cases} H(f, e') - H(f, e^*), & \text{if } \delta(f, e', e^*, W) > 0; \\ 0, & \text{else.} \end{cases}$$

The main technique is the proximity operator for group optimization in line 5, which tries to minimize the function  $Q(\cdot; a, t, \lambda_1, \lambda_2)$  with  $a = W^t + \nabla_W \delta(f, e', e^*, W^t)/t$ :

$$Q(W; a, t, \lambda_1, \lambda_2) = (W - a)^\top W + \frac{2}{t} \left( \lambda_1 \sum_{i=1}^d |W_i| + \lambda_2 \sum_{1 \leq i < j \leq d} \max\{|W_i|, |W_j|\} \right). \quad (3)$$

In the next Section, we will present the details of this proximity for group optimization.

#### 4 Group Optimization

To derive an efficient algorithm with large  $d$  for group optimization, we present the following lemma with its proof attached in appendix.

**Lemma 1.** *In Eq.3, if  $a_k=0$ , then its minimal solution  $\hat{W}$  suffices to  $\hat{W}_k = 0$ .*

Suppose  $W^t$  is sparse, i.e.  $d$  is largely greater than the number of its non-zero components ( $|\Delta(W^t)|$ ), then  $W^{t+1/2}$  in line 4 is also sparse since  $H$  is sparse. The above Lemma states that the optimal solution  $W^{t+1}$  in line 5 of Algorithm 2 is also a sparse vector. Therefore, it is desirable to optimize the  $W^{t+1}$  in a low complexity independent on  $d$ . If so, we can easily see that if we set  $W^1$  as a sparse vector,  $W^t$  is sparse for all  $t > 1$  by a mathematical induction. Based on these analysis, the efficiency of proximity operator for group optimization only requires an assumption that its proximity step can be efficiently solved in a low complexity independent on a large  $d$ .

Let  $u = |\Delta(a)|$ , and  $p$  be a one-to-one map<sup>2</sup>  $p : \{1, \dots, u\} \rightarrow \{1, \dots, d\}$ , s.t.  $|a_{p(1)}| \geq |a_{p(2)}| \geq \dots \geq |a_{p(u)}| > 0$ . Followed by Lemma 1, minimizing Eq.2 is equivalent to minimizing the following equation if we ignore the zero compo-

<sup>2</sup>For easier understanding,  $i'(j')$  denotes the index in  $\{1, \dots, u\}$ , while  $i(j)$  denotes the index in  $\{1, \dots, d\}$ .

nents in the optimal solutions of both equations:

$$\begin{aligned} \bar{Q}(W; a, t, \lambda_1, \lambda_2) &= \sum_{i'=1}^u W_{p(i')}^2 - \sum_{i'=1}^u a_{p(i')} W_{p(i')} \\ &+ \sum_{i'=1}^u \frac{2(\lambda_1 + \lambda_2(d-u))}{t} |W_{p(i')}| + \\ &\frac{2\lambda_2}{t} \sum_{1 \leq i' < j' \leq u} \max\{|W_{p(i')}|, |W_{p(j')}|\}. \end{aligned}$$

The advantage of optimizing  $\bar{Q}(W; a, t, \lambda_1, \lambda_2)$  instead of  $Q(W; a, t, \lambda_1, \lambda_2)$  is that it explicitly reduces the size of active components in  $W$  into  $u$  rather than  $d$ , and thus it is more direct to expect a faster optimization algorithm. Further, Proposition 1 in (Zhong and Kwok, 2011) states that the minimal solution  $\hat{W}$  of such an equation as  $\bar{Q}(W; a, t, \lambda_1, \lambda_2)$  suffices to the constraint  $|\hat{W}_{p(1)}| \geq \dots \geq |\hat{W}_{p(u)}|$ . Therefore, minimizing  $Q(W; a, t, \lambda_1, \lambda_2)$  is also equivalent to optimizing the following constraint programming:

$$\begin{aligned} &\underset{W}{\text{minimize}} && \bar{Q}(W; a, t, \lambda_1, \lambda_2) \\ &\text{subject to} && |W_{p(1)}| \geq \dots \geq |W_{p(u)}|, \end{aligned}$$

where  $\bar{Q}(W; a, t, \lambda_1, \lambda_2)$  defined on the constraint is rewritten as

$$\begin{aligned} \bar{Q}(W; a, t, \lambda_1, \lambda_2) &= \sum_{i'=1}^u W_{p(i')}^2 - \sum_{i'=1}^u a_{p(i')} W_{p(i')} \\ &+ \sum_{i'=1}^u \frac{2(\lambda_1 + \lambda_2(d-i'))}{t} |W_{p(i')}|. \quad (4) \end{aligned}$$

Now, we can implement line 5 in Algorithm 2 as summarized by Algorithm 3, after some modifications over the projection algorithm in (Zhong and Kwok, 2011). Algorithm 3 requires some variables  $\lambda_1, \lambda_2, a$  and  $t$ . Firstly, it sorts  $|a_i|$  for the indice in  $\Delta(a)$  to obtain the map  $p$  in line 1, and initializes  $\mathcal{G}$  as  $\{\{p(1)\}\}$  in line 2. From line 3 to line 10, it goes into a merging loop where it repeatedly merges two group members to pre-calculate  $\mathcal{G}$ : for each  $i'$ , it iteratively merges the member  $g$  initialized as  $\{p(i')\}$  and the top member in the stack, updates  $g$  with the merged member, and substitutes the top member in the stack with  $g$ , if the  $v$  value (will be defined later) of  $g$  is greater than that of the top member. Then, it begins to calculate  $W$  initialized as 0 and  $\mathcal{G}$ . For each index  $i$  in each member  $g$  of  $\mathcal{G}$ , it assigns  $W_i$

---

**Algorithm 3 Group Optimization**

---

**Input:**  $\lambda_1, \lambda_2, a, t$ 

```
1: Sort  $\{|a_i| : i \in \Delta(a)\}$  to obtain  $p \triangleright$  See the definition of  $\Delta$  in Section 2
2: Initialize stack of group set  $\mathcal{G} = \{p(1)\}$ 
3: for all  $i'$  such that  $2 \leq i' \leq |\Delta(a)|$  do
4:    $g = \{p(i')\}$ 
5:   while  $\mathcal{G} \neq \emptyset$  and  $v(g) \geq v(\text{top}(\mathcal{G}))$  do
6:      $g = g \cup \text{top}(\mathcal{G}) \triangleright$  Merge  $g$ 
7:     Pop  $\text{top}(\mathcal{G})$ 
8:   end while
9:   Push  $g$  onto  $\mathcal{G}$ 
10: end for  $\triangleright$  Pre-calculate  $\mathcal{G}$ 
11:  $W = 0$ 
12: for all  $g \in \mathcal{G}$  do
13:   for all  $i \in g$  do
14:      $W_i = \text{sign}(a_i)v(g)$ 
15:   end for
16: end for  $\triangleright$  Calculate  $W$ 
17:  $\mathcal{G} = \mathcal{D}(W)$   $\triangleright$  Calculate  $\mathcal{G}$ 
Output:  $W, \mathcal{G}$   $\triangleright$   $W$  minimizes Eq.3
```

---

according to the sign<sup>3</sup> of  $a_i$  and  $v(g)$  in line 14. In line 17 it calculates  $\mathcal{G} = \mathcal{D}(W)$  as discussed in Section 3. At last it returns the pair  $\langle W, \mathcal{G} \rangle$ .

In particular, the  $v$  value  $v(g)$  in line 5 is defined as

$$v(g) = \frac{\sum_{i \in g} \left( |a_i| - 2(\lambda_1 + \lambda_2(d - p^*(i))) / t \right)}{2|g|},$$

where  $p^*(i)$  denotes the inversion of  $p$  such that  $p(p^*(i)) = i$ . And  $v(g)$  can be intuitively interpreted as the group averaged sub-gradient of  $(\sum_{i'=1}^u W_{p(i')}^2 - \bar{Q}(W; a, t, \lambda_1, \lambda_2)) / 2$ . In addition, an intuitive explanation of merging loop is that the value of objective in Eq.4 will be decreased after each merging step in line 6.

In summary, if we use a sparse representation for vector  $a$  in Algorithm 3, then its complexity is  $O(|\Delta(a)| \log(|\Delta(a)|))$ , which is independent of  $d$ . Therefore, the whole tuning algorithm (Algorithm 1) with feature grouping is efficient even with a large value of  $d$ .

## 5 Experiments

We conduct experiments on the IWSLT2008 Chinese-to-English translation tasks, whose training data consists of about 30K bilingual sentence

<sup>3</sup>The reason is attributed to the Eq.5 in (Zhong and Kwok, 2011).

pairs. Test sets 2003, 2004 and 2008 are used as the development set, development test (devtest) set and test set, respectively; and all of them contain 16 references. A 5-gram language model is trained on the training data with the SRILM toolkit, and word alignment is obtained with GIZA++. In our experiments, the translation performances are measured by the case-insensitive BLEU4 metric. The significance testing is performed by paired bootstrap re-sampling (Koehn, 2004).

We use an in-house developed hierarchical phrase-based translation (Chiang, 2005) as our baseline decoder, and we use the state of the art tuning methods MERT and PRO as our comparison methods<sup>4</sup>. Based on our in-house decoder, we implement three translation models with different feature sets: default features (default); default features plus rule id features (+id); and default features plus group features of rule id (+group). On the IWSLT training data, the number of rule id features is 500K, i.e.  $d = 500K$ , which is significantly greater than the number of bilingual sentences 30K. Our proposed tuning method is with the following setting by tuning on the dev-test set:  $\lambda_1 = 1e - 10$ ,  $\lambda_2 = 3e - 8$ , and  $T = 15$ ,  $n = 20 \times N$ , i.e. 20 passes over k-best-lists.

From Table 1, we can see that tuning the translation model on the development set is much better (improvements of 4.3 BLEU scores) than that on the training data under the default features setting. Its main reason, as presented in Section 1, may be that multiple references and closeness<sup>5</sup> of tuning sets are much helpful for translation tasks. Further, the id features do not achieve improvements and even decreases 0.9 BLEU scores when tuned on the development set, due to its serious sparsity. However, after grouping id features, the groups learned by our method can alleviate the feature sparsity and thus significantly obtain gains of 0.7 BLEU scores over default feature setting.

Further, we implement another tuning method<sup>6</sup> for comparison, i.e.  $L_1$  regularization method (Tsuruoka et al., 2009) based on the ranking loss  $L(W)$  defined in Eq.1. We tune the translation

<sup>4</sup>Both of them are derived from the Moses toolkit: <http://www.statmt.org/moses/>.

<sup>5</sup>If the tuning set and test set are close enough or identically distributed, it is possible to get gains by sparse discriminative features without using feature grouping (Chiang et al., 2009).

<sup>6</sup>It is similar to dtrain implemented in the cdec toolkit: <http://cdec-decoder.org/>, except that it does not use the distributed learning framework.



Methods	Tuning set	Feature set	# Features		BLEU4		Runtimes
			Active	Reused	devtest	test	
MERT	dev	default	8	8	45.7	40.6	15
PRO	dev	default	8	8	46.3	41.1	34
PRO	train	default	8	8	42.8	36.8	834
PRO	dev	+id	11081	4534	45.5	40.2	47
L <sub>1</sub>	train	+id	584	71	42.7	36.9	975
L <sub>1</sub>	dev	+id	443	248	46.2	41.0	39
<b>OSCAR</b>	–	<b>+group</b>	<b>503</b>	<b>425</b>	<b>46.9</b>	<b>41.8</b>	1256

Table 1: BLEU scores on the test set and tuning runtimes (minutes) for the different tuning methods with different settings. Tuning sets dev and train denote the development and training data sets, respectively. "Active" denotes the number of active features for all methods except OSCAR or active grouped features for OSCAR; and "Reused" denotes the number of active (or grouped) features which also appear during 1000-best decoding on the test set. Boldface BLEU means our method OSCAR is significantly better than other methods with  $p < 0.05$ .

model with the +id feature setting on both the development set and training data set, respectively, and their hyperparameters are tuned on the dev-test set. As depicted in Table 1, our method significantly outperforms the L<sub>1</sub> method.

In addition, Table 1 presents the number of both active and reused features for each method on different settings. We can see that the active features (503 grouped features) in OSCAR method are much less than those (11081 features) in PRO with +id setting, which means that OSCAR has lower model complexity. Further, most (84.5%) of active features tuned on dev set are be used during testing for OSCAR, which means that OSCAR is more efficient to address feature sparsity problem compared with both L<sub>1</sub> and PRO.

At last, Table 1 also shows the runtimes for each tuning method. Tuning on training data is much inefficient compared with tuning on dev set, since it requires repeatedly decoding on a much larger dataset. Furthermore, the efficiency of our OSCAR method is comparable to that of tuning on training data. Anyway, distributed training is a reasonable approach to improve the efficiency of OSCAR, as suggested by Simianer et al. (2012).

## 6 Conclusion and Future Work

This paper proposes a novel training method for a translation model with a large number of features, which is the main contribution of this paper. This method is based on automatic feature grouping, which is implemented within an online learning method and thus is efficient for large scale training in SMT. The other contribution is that we success-

fully extend OSCAR to a large scale of learning setting. In future work, we will investigate distributed learning for OSCAR and then testify it on larger scale training data.

## Acknowledgments

We would like to thank our colleagues in both HIT and NICT for insightful discussions, and three anonymous reviewers for many invaluable comments and suggestions to improve our paper. This work is supported by National Natural Science Foundation of China (61173073, 61100093, 61073130, 61272384), and the Key Project of the National High Technology Research and Development Program of China (2011AA01A207).

## References

- H. D. Bondell and B. J. Reich. 2008. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123.
- David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. of EMNLP. ACL*.
- David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL, NAACL '09*, pages 218–226.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL, ACL '05*, pages 263–270.
- John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *J. Mach. Learn. Res.*, 10:2899–2934, December.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *EMNLP*, pages 1352–1362, July.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP. ACL*.

Mu Li, Yinggong Zhao, Dongdong Zhang, and Ming Zhou. 2010. Adaptive development data selection for log-linear model in statistical machine translation. In *COLING, COLING '10*, pages 662–670.

J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. of 5-th Berkeley Symposium on Mathematical Statistics and Probability*.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *ACL, ACL '12*, pages 11–21.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *ACL-IJCNLP, ACL '09*, pages 477–485.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proc. of EMNLP-CoNLL*.

Xinyan Xiao, Yang Liu, Qun Liu, and Shouxun Lin. 2011. Fast generation of translation forest for large-scale smt discriminative training. In *EMNLP*, pages 880–888.

Wenliang Zhong and James Kwok. 2011. Efficient sparse modeling with automatic feature grouping. In *ICML, ICML '11*, pages 9–16.

## Appendix

*Proof.* Suppose  $\hat{W}_k \neq 0$ , and thus  $|\hat{W}_k| > 0$ . Set  $\hat{W}'$  as another weight such that  $\hat{W}'_j = \hat{W}_j$  for all  $j (j \neq k)$ , and  $\hat{W}'_k = 0$ . Then, for each  $i, j$  the following equations hold:

$$|\hat{W}_i| \geq |\hat{W}'_i|,$$

and

$$\max\{|\hat{W}_i|, |\hat{W}_j|\} \geq \max\{|\hat{W}'_i|, |\hat{W}'_j|\}.$$

Thus, the following equations hold based on the above equations by simple algebraic operations:

$$\begin{aligned} & Q(\hat{W}; a, t, \lambda_1, \lambda_2) - Q(\hat{W}'; a, t, \lambda_1, \lambda_2) \\ &= \hat{W}_k \times \hat{W}_k + \frac{2\lambda_1}{t} \sum_{i=1}^d (|\hat{W}_i| - |\hat{W}'_i|) + \frac{2\lambda_2}{t} \times \\ & \quad \sum_{1 \leq i < j \leq d} (\max\{|\hat{W}_i|, |\hat{W}_j|\} - \max\{|\hat{W}'_i|, |\hat{W}'_j|\}) \\ & \geq \hat{W}_k \times \hat{W}_k > 0. \end{aligned}$$

Therefore, we conclude that  $Q(\hat{W}; a, t, \lambda_1, \lambda_2) > Q(\hat{W}'; a, t, \lambda_1, \lambda_2)$ . This contradicts the assumption that  $\hat{W}$  is the minimal solution of Eq.3.  $\square$

# Multimodal Comparable Corpora as Resources for Extracting Parallel Data: Parallel Phrases Extraction

Haithem Affi, Loïc Barrault and Holger Schwenk

Université du Maine,

Avenue Olivier Messiaen F-72085 - LE MANS, France

FirstName.LastName@lium.univ-lemans.fr

## Abstract

Discovering parallel data in comparable corpora is a promising approach for overcoming the lack of parallel texts in statistical machine translation and other NLP applications. In this paper we propose an alternative to comparable corpora of texts as resources for extracting parallel data: a multimodal comparable corpus of audio and texts. We present a novel method to detect parallel phrases from such corpora based on splitting comparable sentences into fragments, called phrases. The audio is transcribed by an automatic speech recognition system, split into fragments and translated with a baseline statistical machine translation system. We then use information retrieval in a large text corpus in the target language, split also into fragments, and extract parallel phrases. We compared our method with parallel sentences extraction techniques. We evaluate the quality of the extracted data on an English to French translation task and show significant improvements over a state-of-the-art baseline.

## 1 Introduction

The development of a statistical machine translation (SMT) system requires one or more parallel corpora called bitexts for training the translation model and monolingual data to build the target language model. Unfortunately, parallel texts are a limited resource and they are often not available for some specific domains and language pairs. That is why, recently, there has been a huge interest in the automatic creation of parallel data. Since comparable corpora exist in large quantities and are much more easily available (Munteanu and Marcu, 2005), the ability to exploit them is highly

beneficial in order to overcome the lack of parallel data. The ability to detect these parallel data enables the automatic creation of large parallel corpora.

Most of existing studies dealing with comparable corpora look for parallel data at the sentence level (Zhao and Vogel, 2002; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005; Abdul-Rauf and Schwenk, 2011). However, the degree of parallelism can vary considerably, from noisy parallel texts, to quasi parallel texts (Fung and Cheung, 2004). Corpora from the last category contain none or few good parallel sentence pairs. However, there could have parallel phrases in comparable sentences that can prove to be helpful for SMT (Munteanu and Marcu, 2006). As an example, consider Figure 1, which presents two news articles with their video from the English and French editions of the Euronews website<sup>1</sup>. The articles report on the same event with different sentences that contain some parallel translations at the phrase level. These two documents contain in particular no exact sentence pairs, so techniques for extracting parallel sentences will not give good results. We need a method to extract parallel phrases which exist at the sub-sentential level.

For some languages, text comparable corpora may not cover all topics in some specific domains and languages. This is because potential sources of comparable corpora are mainly derived from multilingual news reporting agencies like AFP, Xinhua, Al-Jazeera, BBC etc, or multilingual encyclopedias like Wikipedia, Encarta etc. What we need is exploring other sources like audio to generate parallel data for such domains that can improve the performance of an SMT system.

In this paper, we present a method for detecting and extracting parallel data from multimodal corpora. Our method consists in extracting parallel

---

<sup>1</sup>[www.euronews.com/](http://www.euronews.com/)

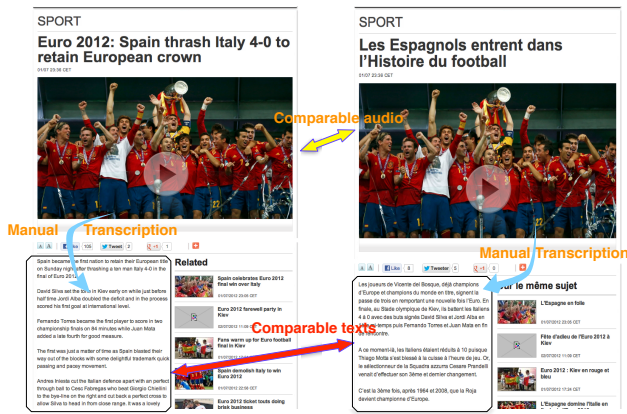


Figure 1: Example of multimodal comparable corpora from the Euronews website.

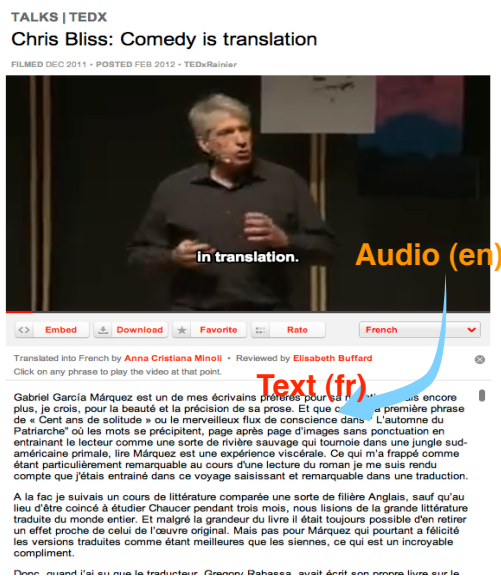


Figure 2: Example of multimodal comparable corpora from the TED website.

phrases.

## 2 Extracting parallel data

### 2.1 Basic Idea

Figure 2 shows an example of multimodal comparable data coming from the TED website<sup>2</sup>. We have an audio source of a talk in English and its text translation in French. We think that we can extract parallel data from this corpora, at the sentence and the sub-sentential level.

In this work we seek to adapt and to improve machine translation systems that suffer from resource deficiency by automatically extracting parallel data in specific domains.

<sup>2</sup><http://www.ted.com/>

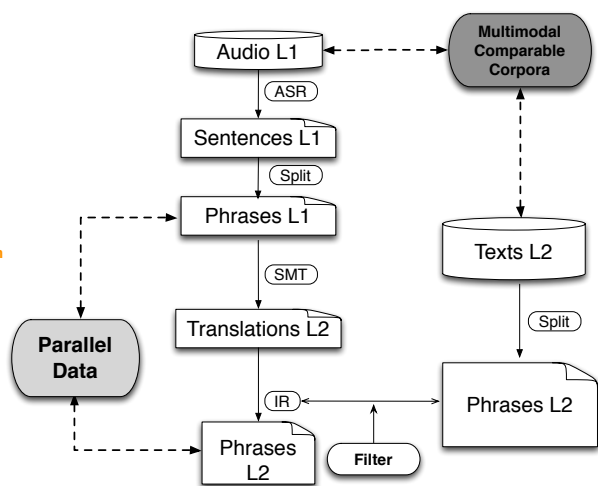


Figure 3: Principle of the parallel phrase extraction system from multimodal comparable corpora.

### 2.2 System Architecture

The basic system architecture is described in Figure 3. We can distinguish three steps: automatic speech recognition (ASR), statistical machine translation (SMT) and information retrieval (IR). The ASR system accepts audio data in the source language L1 and generates an automatic transcription. This transcription is then split into phrases and translated by a baseline SMT system into language L2. Then, we use these translations as queries for an IR system to retrieve most similar phrases in the texts in L2, which were previously split into phrases. The transcribed phrases in L1 and the IR result in L2 form the final parallel data. We hope that the errors made by the ASR and SMT systems will not impact too severely the extraction process.

Our technique is similar to that of (Munteanu and Marcu, 2006), but we bypass the need of the Log-Likelihood-Ratio lexicon by using a baseline SMT system and the TER measure (Snover et al., 2006) for filtering. We also report an extension of the work of (Aflî et al., 2012) by splitting transcribed sentences and the text parts of the multimodal corpus into phrases with length between two to ten tokens. We extract from each sentence on the corpus all combinations of two to ten sequential words.

## 2.3 Baseline systems

Our ASR system is a five-pass system based on the open-source CMU Sphinx toolkit<sup>3</sup>(version 3 and 4), similar to the LIUM'08 French ASR system described in (Deléglise et al., 2009). The acoustic models are trained in the same manner, except that a multi-layer perceptron (MLP) is added using the bottle-neck feature extraction as described in (Grézl and Fousek, 2008). Table 2.3 shows the performances of the ASR system on the development and test corpora.

Corpus	% WER
Development	19.2
Test	17.4

Table 1: Performance of the ASR system on development and test data.

Our SMT system is a phrase-based system (Koehn et al., 2003) based on the Moses SMT toolkit (Koehn et al., 2007). The standard fourteen feature functions are used, namely phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, a word and a phrase penalty and a target language model. It is constructed as follows. First, word alignments in both directions are calculated. We used the multi-threaded version of the GIZA++ tool (Gao and Vogel, 2008). Phrases and lexical reorderings are extracted using the default settings of the Moses toolkit. The parameters of our system were tuned on a development corpus, using the MERT tool (Och, 2003).

We use the Lemur IR toolkit (Ogilvie and Callan, 2001) for the phrases extraction procedure. We first index all the French text (after splitting it into segments) into a database using *Indri Index*. This feature enable us to index our text documents in such a way we can use the translated phrases as queries to run information retrieval in the database, with the specialized *Indri Query Language*. By these means we can retrieve the best matching phrases from the French side of the comparable corpus.

For each candidate phrases pair, we need to decide whether the two phrases are mutual translations. For this, we calculate the TER between them using the tool described in (Servan and

Schwenk, 2011),<sup>4</sup> i.e. between automatic translation, and the phrases selected by IR.

## 3 Experiments

In our experiments, we compare our phrase extraction method (which we call *PhrExtract*) with the sentence extraction method (*SentExtract*) of (Afli et al., 2012). We use the extracted dataset by both methods as additional SMT training data, and measure the quality of the parallel data by its impact on the performance of the SMT system. Thus, the final extracated parallel data is injected into the baseline system. The various SMT systems are evaluated using the BLEU score (Papineni et al., 2002). We conducted experiments on an English to French machine translation task. All the text data is automatically split into phrases of two to ten tokens.

### 3.1 Data description

Our multimodal comparable corpus consists of spoken talks in English (audio) and written texts in French. The goal of the TED task is to translate public lectures from English into French. The TED corpus totals about 118 hours of speech. We call the English transcriptions of the audio part *TEDasr* witch is split into phrases (called *TEDasr\_split*). A detailed description of the TED task can be found in (Rousseau et al., 2011).

The development corpus *DevTED* consists of 19 talks and represents a total of 4 hours and 13 minutes of speech transcribed at the sentence level. The language model is trained with the SRI LM toolkit (Stolcke, 2002), on all the available French data without the TED data. The baseline system is trained with version 7 of the News-Commentary (nc7) and Europarl (eparl7) corpus.<sup>5</sup> The indexed data consist of the French text part of the *TED* corpus which contains translations of the English part of the corpus. We call it *TEDbi*. It is split into phrases (called *TEDbi\_split*). Tables 2 and 3 summarize the characteristics of the different corpora used in our experiments.

### 3.2 Experimental results

We first apply sentence extraction on the TED corpus with a method similar to (Afli et al., 2012). We then apply phrase extraction on the same data split

<sup>3</sup>Carnegie Mellon University:  
<http://cmusphinx.sourceforge.net/>

<sup>4</sup><http://sourceforge.net/projects/tercpp/>

<sup>5</sup><http://www.statmt.org/europarl/>

bitexts	# tokens	in-domain ?
nc7	3.7M	no
eparl7	56.4M	no
DevTED	36k	yes

Table 2: MT training and development data.

Data	# tokens	in-domain ?
TEDasr	1.8M	yes
TEDbi	1.9M	yes
TEDbi_split	80.4M	yes
TEDasr_split	82.7M	yes

Table 3: Comparable data used for the extraction experiments.

as described in 2.2. Then, both methods are compared.

As mentioned in section 2.3, the TER score is used as a metric for filtering the result of IR. We keep only the sentences or phrases which have a TER score below a certain threshold determined empirically. Thus, we filter the selected sentences or phrases in each condition with different TER thresholds ranging from 0 to 100 by steps of 10. The extracted parallel data are added to our generic training data in order to adapt the baseline system. Table 4 presents the BLEU score obtained for these different experimental conditions.

Our baseline SMT system, trained with generic bitexts achieves a BLEU score of 22.93. We can see that our new method of phrase extraction significantly improve the baseline system more than sentences extraction method until the TER threshold of 80 is reached: the BLEU score increases from 22.93 to 23.70 with the best system of our proposed method and from 22.93 to 23.40 with the best system using the classical method of sentence extraction.

The results show that the choice of the appropriate TER threshold depends on the method. We can see that for *PhrExtract* the best threshold is 60 when the best one is 80 for *SentExtract*. This last one is also an important point in the general evaluation of the two methods. In fact, we can see on Figure 4 that from this point our proposed method gives less performing results than *SentExtract* method.

This suggest to apply combination of the two methods. This corresponds to injecting the extracted phrases and sentences into the training

data. The combination method is called *CombExtract*. Figure 4 presents the comparison of the different experimental conditions in term of BLEU score for each TER threshold. We can see that except for threshold 30, the curve of the combination follows in general the same trajectory of the curve of *PhrExtract*. These results show that *SentExtract* has no big impact in combination with the *PhrExtract* method and the best threshold when using *PhrExtract* is at 60.

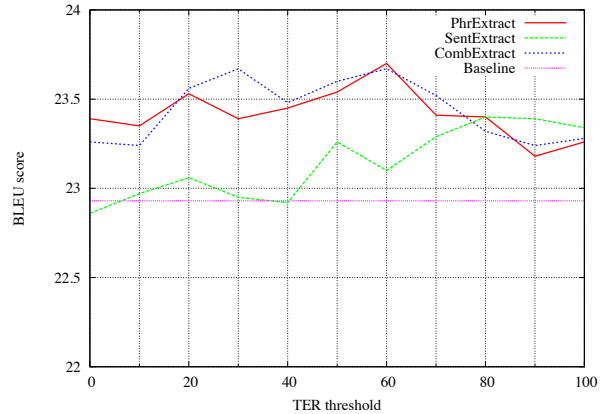


Figure 4: Performance of *PhrExtract*, *SentExtract* and their combination in term of BLEU score for each TER threshold.

This is because of the big difference on the quantity of data between the two methods as we can see in Table 4. The benefit of our method is that it can generate more quantities of parallel data than the sentence extraction method for each TER threshold, and this difference of quantities improves results of MT system until the TER threshold of 80 is reached. However, we can see in Table 4 that the quality of only 39.35k (TER 80) extracted by *SentExtract* can have exactly the same impact of 25.3M extracted by our new technique. That is why we intend to investigate in the filtering module of our system.

## 4 Related Work

Research on exploiting comparable corpora goes back to more than 15 years ago (Fung and Yee, 1998; Koehn and Knight, 2000; Vogel, 2003; Gaussier et al., 2004; Li and Gaussier, 2010). A lot of studies on data acquisition from comparable corpora for machine translation have been reported (Su and Babych, 2012; Hewavitharana and Vogel, 2011; Riesa and Marcu, 2012).

To the best of our knowledge (Munteanu and

TER	BLEU score SentExtract	BLEU score PhrExtract	# tokens (fr) SentExtract	# tokens (fr) PhrExtract
0	22.86	23.39	55	1.06M
10	22.97	23.35	313	1.4M
20	23.06	23.53	1.7k	2.5M
30	22.95	23.39	6.9k	4.3M
40	22.92	23.45	23.5k	7.02M
50	23.26	23.54	62.4k	11.4M
60	23.10	<b>23.70</b>	13.82k	<b>13.8M</b>
70	23.29	23.41	25.15k	18.04M
80	<b>23.40</b>	<b>23.40</b>	<b>39.35k</b>	<b>25.3M</b>
90	23.39	23.18	57.54k	35.9M
100	23.34	23.26	83.60k	45.3M
Baseline	22.93	-	60.1M	-

Table 4: Number of tokens extracted and BLEU scores on DevTED obtained with *PhrExtract* and *SentExtract* methods for each TER threshold.

Marcu, 2006) was the first attempt to extract parallel sub-sentential fragments (phrases), from comparable corpora. They used a method based on a Log-Likelihood-Ratio lexicon and a smoothing filter. They showed the effectiveness of their method to improve an SMT system from a collection of a comparable sentences. The weakness of their method is that they filter source and target fragments separately, which cannot guarantee that the extracted fragments are a good translations of each other. (Hewavitharana and Vogel, 2011) show a good result with their method based on on a pairwise correlation calculation which suppose that the source fragment has been detected.

The second type of approach in extracting parallel phrases is the alignment-based approach (Quirk et al., 2007; Riesa and Marcu, 2012). These methods are promising, but since the proposed method in (Quirk et al., 2007) do not improve significantly MT performance and model in (Riesa and Marcu, 2012) is designed for parallel data, it’s hard to say that this approach is actually effective for comparable data.

This work is similar to the work by (Afli et al., 2012) where the extraction is done at the phrase level instead of the sentence level. Our methodology is the first effort aimed at detecting translated phrases on a multimodal corpora.

Since our method can extract parallel phrases from a multimodal corpus, it greatly expands the range of corpora which can be usefully exploited.

## 5 Conclusion

We have presented a fully automatic method for extracting parallel phrases from multimodal comparable corpora, *i.e.* the source side is available as audio stream and the target side as text. We used a framework to extract parallel data witch combine an automatic speech recognition system, a statistical machine translation system and information retrieval system. We showed by experiments conducted on English-French data, that parallel phrases extracted with this method improves significantly SMT performance. Our approach can be improved in several aspects. The automatic splitting is very simple; more advanced phrases generation might work better, and eliminate redundancy. Trying other method on filtering can also improve the precision of the method.

## 6 Acknowledgments

This work has been partially funded by the French Government under the project DEPART.

## References

- S. Abdul-Rauf and H. Schwenk. 2011. Parallel sentence generation from comparable corpora for improved smt. *Machine Translation*.
- H. Afli, L. Barrault, and H. Schwenk. 2012. Parallel texts extraction from multimodal comparable corpora. In *JapTAL*, volume 7614 of *Lecture Notes in Computer Science*, pages 40–51. Springer.
- P. Deléglise, Y. Estève, S. Meignier, and T. Merlin. 2009. Improvements to the LIUM french ASR sys-

- tem based on CMU Sphinx: what helps to significantly reduce the word error rate? In *Interspeech 2009*, Brighton (United Kingdom), 6-10 september.
- P. Fung and P. Cheung. 2004. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04.
- P. Fung and L. Y. Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pages 414–420.
- Q. Gao and S. Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57.
- E. Gaussier, J.-M. Renders, I. Matveeva, C. Goutte, and H. Déjean. 2004. A geometric view on bilingual lexicon extraction from comparable corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04.
- F. Grézl and P. Fousek. 2008. Optimizing bottle-neck features for LVCSR. In *2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4729–4732. IEEE Signal Processing Society.
- S. Hewavitharana and S. Vogel. 2011. Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, BUCC '11, pages 61–68.
- P. Koehn and K. Knight. 2000. Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 711–715. AAAI Press.
- P. Koehn, Franz J. Och, and D. 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.
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. 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.
- B. Li and E. Gaussier. 2010. Improving corpus comparability for bilingual lexicon extraction from comparable corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 644–652.
- D. S. Munteanu and D. Marcu. 2005. Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Computational Linguistics*, 31(4):477–504.
- D. S. Munteanu and D. Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 81–88.
- Franz J. 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.
- P. Ogilvie and J. Callan. 2001. Experiments using the lemur toolkit. *Proceeding of the Tenth Text Retrieval Conference (TREC-10)*.
- K. Papineni, S. Roukos, T. Ward, and W.-J. 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.
- Q. Quirk, R. Udupa, and A. Menezes. 2007. Generative models of noisy translations with applications to parallel fragment extraction. In *In Proceedings of MT Summit XI, European Association for Machine Translation*.
- J. Riesa and D. Marcu. 2012. Automatic parallel fragment extraction from noisy data. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 538–542.
- A. Rousseau, F. Bougares, P. Deléglise, H. Schwenk, and Y. Estève. 2011. LIUM's systems for the IWSLT 2011 speech translation tasks. *International Workshop on Spoken Language Translation 2011*.
- C. Servan and H. Schwenk. 2011. Optimising multiple metrics with mert. *The Prague Bulletin of Mathematical Linguistics (PBML)*.
- S. Snover, B. Dorr, R. Schwartz, M. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.
- A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *International Conference on Spoken Language Processing*, pages 257–286, November.



- F. Su and B. Babych. 2012. Measuring comparability of documents in non-parallel corpora for efficient extraction of (semi-)parallel translation equivalents. In *Proceedings of the Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, EACL 2012, pages 10–19. Association for Computational Linguistics.
- M. Utiyama and H. Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 72–79.
- S. Vogel. 2003. Using noisy bilingual data for statistical machine translation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 2*, EACL '03, pages 175–178.
- B. Zhao and S. Vogel. 2002. Adaptive parallel sentences mining from web bilingual news collection. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, Washington, DC, USA. IEEE Computer Society.

# Bootstrapping Large-scale Named Entities using URL-Text Hybrid Patterns

Chao Zhang      Shiqi Zhao      Haifeng Wang

Baidu Inc

No. 10, Shangdi 10th Street, Haidian District, Beijing 100085, China

{zhangchao01, zhaoshiqi, wanghaifeng}@baidu.com

## Abstract

Automatically mining named entities (NE) is an important but challenging task, pattern-based and bootstrapping strategy is the most widely accepted solution. In this paper, we propose a novel method for NE mining using web document titles. In addition to the traditional text patterns, we propose to use url-text hybrid patterns that introduce url criterion to better pinpoint high-quality NEs. We also design a multiclass collaborative learning mechanism in bootstrapping, in which different patterns and different classes work together to determine better patterns and NE instances. Experimental results show that the precision of NEs mined with the proposed method is 0.96 and 0.94 on Chinese and English corpora, respectively. Comparison result also shows that the proposed method significantly outperforms a representative method that mines NEs from large-scale query logs.

## 1 Introduction

The task of named entity mining (NEM) aims to mine named entities (NE) of given categories from raw data. NEM is essential in many applications. For example, NEM can generate NE gazetteers necessary for the task of named entity recognition (NER) (Cohen and Sarawagi, 2004; Kazama and Torisawa, 2008; Talukdar et al., 2006). It can also help improve the search results in web search (Paşca, 2004), and increase the coverage of knowledge graphs.

Extensive research has been conducted on NEM, in which pattern-based methods are the most popular. Handcrafted or automatically learnt patterns are usually used to extract NE instances from various corpora, such as web documents,

search engine's retrieved snippets, and query logs. Bootstrapping strategy is often applied to generate more patterns and instances iteratively so as to improve the coverage of the system.

The method we propose also belongs to the family of pattern-based NEM. However, our method is a departure from the previous ones. It makes contributions from the following aspects: First, we design url-text hybrid patterns instead of the traditional text patterns. We take url criterion into account, so as to measure the quality of the source webpages. Second, we propose Multiclass Collaborative Learning (MCL) mechanism, which globally scores and ranks the patterns and NE instances within multiple classes in bootstrapping.

We evaluate our method in two languages, i.e., Chinese and English, so as to demonstrate the language-independent nature of the method. We mine NEs using the system for five categories in both languages, including *star*, *film*, *TV play*, *song*, and *PC game*. Experimental results show that the average precision of the extracted NEs is 96% in Chinese and 94% in English. Meanwhile, the average coverage computed against a benchmark repository is 61% and 55% for the two languages. Comparative experiments further show that our method significantly outperforms a representative conventional method.

## 2 Related Work

In this section, we review the previous studies on NEM from three aspects: the data resource used, the proposed methods, and particularly the bootstrapping strategy.

Various resources have been exploited for NEM. Many researchers make use of large-scale web corpora and learn NEs surrounded by certain context patterns (Paşca, 2004; Downey et al., 2007). Others mine NEs using web search engines. They submit extraction patterns as queries to search engines, and extract NEs matching the

patterns from the retrieved snippets (Etzioni et al., 2005; Etzioni et al., 2004; ?; Kozareva and Hovy, 2010). There are also studies extracting NEs structured HTML tables (Dalvi et al., 2012). Besides web documents, NEs as well as their attributes can also be mined from search engine query logs, since many users tend to search for named entities in their queries (Paşca, 2007a; Paşca, 2007b).

As an alternative, this paper proposes to mine NEs from vertical websites titles, based on our observation that NEs of a class  $c$  can generally be found in webpage titles of some vertical websites of class  $c$ . Our statistics show that 99% out of 10,000 random NEs appear in webpage titles. Besides, webpage titles have the advantage that they are of better quality than free-text documents, while less noisy than user queries.

Pattern-based methods are the most popular ones in NEM (Riloff and Jones, 1999; Thelen and Riloff, 2002; Etzioni et al., 2004; Paşca, 2004; Talukdar et al., 2006; Paşca, 2007b; Wang and Cohen, 2009; Kozareva and Hovy, 2010). NE extraction patterns in previous papers can be roughly classified into two types, i.e., Hearst patterns and class-specific wrappers. Hearst patterns are named after Hearst (1992), who among the first to design patterns, such as “E is a C”, “C including E”, to extract hyponyms / hypernyms. The surface patterns were later extended to lexico-syntactic patterns (Thelen and Riloff, 2002; Paşca, 2004), so that the pattern-filling instances can be identified more accurately via considered constraints.

Hearst patterns are binary patterns containing two slots. In contrast, class-specific wrappers are unary patterns with a single slot (Paşca, 2007b; Wang and Cohen, 2008). For example, the pattern “the film \* was directed by” is a wrapper for the *film* class, in which the place holder “\*” can be replaced by any film name. Wrappers need to be learnt for each NE class of interest. Our method proposed in this paper is also a pattern-based one. However, we design a novel type of url-text hybrid pattern, which not only benefits from the conventional textual wrappers, but also takes advantage of url constraint.

Most methods mentioned above are weakly-supervised, in which a few patterns, heuristic rules or instances are fed to the system as seeds, and the system enriches patterns and NE instances iteratively. Bootstrapping is widely used in these methods (Riloff and Jones, 1999; Thelen and Riloff-

f, 2002; Paşca, 2007b; Wang and Cohen, 2008). The bootstrapping algorithm can effectively reduce manual intervention in building the system. However, it is prone to noise brought in during iterations. We therefore design a Multiclass Collaborative Learning (MCL, detailed in Section 3.4) mechanism in this paper, which guarantees the quality of the generated new patterns and instances by introducing inter-class and intra-class scoring criteria.

### 3 Named Entity Mining

#### 3.1 Overview of the Method

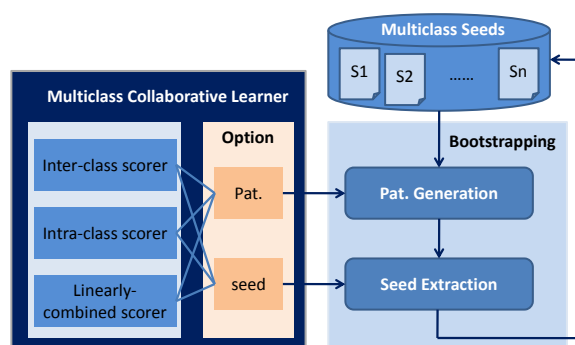


Figure 1: Framework of named entity mining

Named entities of a category are often organized in corresponding vertical websites, where each named entity is displayed in a single webpage. For example, it’s easy to extract film NEs from IMDB<sup>1</sup> web titles with regular expressions.

Our method learns NE extraction patterns from web titles (text pattern) and introduces url constraint (url pattern) to make the extraction results more precise. As described in Figure 1, our method uses bootstrapping strategy in pattern generation and seed extraction. We also propose Multiclass Collaborative Learning (MCL) mechanism to filter noise introduced in iterations.

#### 3.2 URL-Text Hybrid Patterns

##### 3.2.1 Motivation

Text patterns are widely used as wrappers in tasks like information extraction and relation extraction. To improve the accuracy of wrappers, a lot of constraints such as part-of-speech tags (Etzioni et al., 2005) and trigger words (Talukdar et al., 2006) were introduced to tackle the

<sup>1</sup>www.imdb.com

tricky conditions. However, simple wrappers can also acquire high-quality NEs in specific conditions. For example, “ $^(.+?)$$ ” is qualified to extract person name from the titles whose corresponding urls match the regular expression “ $http://www.nndb.com/people/\d+/\d+/$$ ”.

Based on the consideration above, we take the quality of urls into consideration when using wrappers. We use simple text patterns if the websites are of high-quality, and have to use complicated text patterns if the website’s quality is low. We therefore design url-text hybrid patterns to guarantee the capability of the patterns from both url and text aspects.

### 3.2.2 URL Patterns

Similar urls share the same pattern in many websites (Blanco et al., 2011). For example, all IMDB webpages describing video information match pattern “ $http://www.imdb.com/title/tt\d+/$$ ”. Therefore, we can take the url pattern as the identity of the website. Url patterns are globally learned using a large-scale url database. The process is as follows:

1. Given a url, we generate candidate url patterns by replacing the segments separated with “/” from its non-domain parts with slots respectively. For example, for url: “ $www.AAA.com/BBB/123$ ” in which “ $/BBB/123$ ” is the non-domain part, we can generate two candidate patterns “ $www.AAA.com/SLOT/123$ ” and “ $www.AAA.com/BBB/SLOT$ ”.
2. All candidate patterns are accumulated on the url database. The ones with a frequency above a pre-defined threshold  $k$  are retained.
3. For each retained candidate pattern, we generate the final url pattern by replacing the slot with a regular expression based on the statistics of its slot fillers. For instance, for the candidate pattern “ $www.AAA.com/BBB/SLOT$ ”, if the slot can be filled with “123”, “234”, and “456”, then the slot can be replaced with “ $\d+$ ”, meaning that this slot can be filled with any number sequence. Accordingly, the final url pattern should be “ $www.AAA.com/BBB/\d+$ ”.

### 3.2.3 Text Patterns

Text patterns are commonly used in NEM. Here we use a classical method to generate text patterns. Given a seed NE  $s$  in a category  $c$ , and a title  $t$  containing  $s$ , the text patterns are generated as:

1. Segment the title  $t$  into a word sequence.
2. Match the seed  $s$  in  $t$ , and replace  $s$  with the slot “ $^(.+?)$$ ”<sup>2</sup>.
3. Generate patterns that contain the slot as well as words preceding and succeeding the slot within a pre-defined window size. Several patterns can be yielded in this way given different window sizes. We set the window size to 2, 3, 4, and 5 in our experiment.

### 3.2.4 Hybrid Patterns

A url-text hybrid pattern ( $utp$ ), combining both url and text patterns, is defined as a 4-tuple:  $utp = (up, tp, c, f)$ , where  $up$  and  $tp$  are the url pattern and text pattern respectively,  $c$  indicates the category that  $utp$  belongs to, and  $f$  (scored by Eq.(6)) denotes the confidence of  $utp$ .

We use  $UTP$  to denote a set of  $utps$ , and use  $UTP_i$  to denote the  $UTP$  of the  $i$ -th category  $c_i$ . A hybrid pattern is more strict than a url pattern and a text pattern separately. As we will show, the NEs it can extract are of better quality and coverage.

## 3.3 Bootstrapping

As described in Algorithm 1, our method *GenerateUTP* generates raw patterns ( $r\_UTP^k$ ) with seeds ( $Seeds^{k-1}$ ) from web titles (WT) in the  $k$ -th iteration. Likewise, raw NE instances ( $r\_ins^k$ ) are extracted by *ExtractNE* in the following steps. *SelectUTP* and *SelectNE* output high quality patterns and NE instances respectively. These two functions are based on MCL mechanism described in Section 3.4. During these processes, each pattern is scored by Eq.(5) and is kept if its score is above a threshold, and the instances yielded in each iteration are ranked according to Eq.(6), and those ranked in the top  $1/k$  are selected and added (with function *AddSeeds*) into the seed set.

We use  $\#(ins^k)$  to denote the number of instances after the  $k$ -th iteration. The iterations will terminate if  $\#(ins^k)/\#(ins^{k-1}) < \eta$ , where  $\eta$  is

<sup>2</sup>“ $^(.+?)$$ ” is a regular expression used to extract arbitrary strings

---

**Algorithm 1** Bootstrapping for NE Mining

---

**Require:**

$Seeds^0$  for  $n$  categories:  $\{S_1^0, S_2^0, \dots, S_n^0\}$   
webpage titles (WT);  
iteration count  $k = 1$ ;

**Ensure:**

- 1: **while** Terminate criterion is not met **do**
  - 2:  $r\_UTP^k = GenerateUTP(Seeds^{k-1}, WT)$ ;
  - 3:  $UTP^k =$   
 $SelectUTP(r\_UTP^k, Seeds^{k-1}, WT)$ ;
  - 4:  $r\_ins^k = ExtractNE(UTP^k, WT)$ ;
  - 5:  $ins^k = SelectNE(r\_ins^k)$ ;
  - 6:  $Seeds^k = AddSeeds(ins^k, Seeds^{k-1})$ ;
  - 7:  $k = k + 1$ ;
  - 8: **end while**
  - 9: **return**  $ins^k$ : Named entities for  $n$  categories  
 $\{NE_1, NE_2, \dots, NE_n\}$ ;
- 

a threshold ( $\eta = 1.01$  in our experiments). All the extracted NEs after the last iteration are output along with their confidence score computed according to Equ.(6). One can set threshold w.r.t. the confidence score, so as to select high-quality named entities for certain applications.

### 3.4 Multiclass Collaborative Learning (MCL)

In this section, we design collaborative learning mechanism, which contains inter-class and intra-class scoring criteria, to better control the quality of the patterns and NE instances bootstrapped in iterations.

#### 3.4.1 Inter-class Scoring

If an NE of category  $c_i$  can also be extracted with patterns from other categories, it is likely that it is noise, or at least is an ambiguous NE that is unsuitable to be used as a seed of  $c_i$ . Likewise, if a pattern of class  $c_i$  can also be generated by seeds from other categories, this pattern is obviously not a high-quality pattern for category  $c_i$ . Thus, we can score the patterns and seeds of the target category with the help of the other classes, which is termed “inter-class scoring”.

The inter-class score for patterns is defined as:

$$P_1(c_i|utp) = \frac{P(c_i) \times P(utp|c_i)}{\sum_j P(c_j) \times P(utp|c_j)} \quad (1)$$

where:  $P(c_i) = |S_i|/|\bigcup S_j|$ , in which  $S_i$  denotes the seed set of category  $c_i$  and  $|\cdot|$  means the size

of a set. During initialization, we prepare approximately the same number of seeds for each class.  $P(utp|c_i) = |S_i(utp)|/|S_i|$ , in which  $S_i(utp)$  denotes the set of seeds in class  $c_i$  which generate the pattern  $utp$ .

The inter-class score for instances<sup>3</sup> is defined as:

$$P_1(c_i|s) = \frac{P(c_i) \times P(s|c_i)}{\sum_j P(c_j) \times P(s|c_j)} \quad (2)$$

where:  $P(c_i)$  is defined as above, and  $P(s|c_i) = \frac{Freq_i(s)}{\sum_{s' \in S_i} Freq_i(s')}$ ,  $Freq_i(s)$  means the number of  $c_i$ 's patterns that can extract instance  $s$ ,  $S_i$  means all instances of category  $c_i$ .

#### 3.4.2 Intra-class Scoring

Besides inter-class scoring, we also design an intra-class scoring criterion. The basic hypothesis is that, if a pattern generates a lot of instances that cannot be recalled by other patterns in this class, the pattern is likely to be incorrect.

For each class  $c_i$ , and the set of  $m$  patterns in the current iteration  $UTP_i = utp_i^1, utp_i^2, \dots, utp_i^m$ , we compute the intra-class score for  $utp$  (say,  $utp$  is the  $j$ -th pattern  $utp_i^j$ ) as:

$$P_2(c_i|utp) = \frac{|S_i(utp) \cap S_i^H|}{|S_i(utp)|} \quad (3)$$

where  $S_i(utp)$  means the set of instances extracted by  $utp$  in class  $c_i$ ,  $S_i^H$  is a set of high-quality instances extracted with all patterns in class  $c_i$ . Here “high-quality” is guaranteed by discarding the instances with frequency lower than a threshold  $T$ .

Likewise, intra-class scoring can also be defined for instances: the instances matching more patterns in class  $c_i$  are more likely to be correct instances of this class. The intra-class score for a seed  $s$  is computed as:

$$P_2(c_i|s) = \frac{|UTP_i(s)|}{|UTP_i|} \quad (4)$$

where  $UTP_i(s)$  denotes the set of patterns in class  $c_i$  that can extract instance  $s$ , while  $UTP_i$  denotes the set of all patterns in  $c_i$ .

#### 3.4.3 Linearly-combined Scoring

The final score for patterns and instances linearly combines both inter- and intra-class scores as follows:

---

<sup>3</sup>we also use  $s$  to denote an instance generated during bootstrapping.

For patterns:

$$P(c_i|utp) = \lambda P_1(c_i|utp) + (1 - \lambda)P_2(c_i|utp) \quad (5)$$

For instances:

$$P(c_i|s) = \lambda P_1(c_i|s) + (1 - \lambda)P_2(c_i|s) \quad (6)$$

In our experiments,  $\lambda$  is set to 0.5.

## 4 Evaluation

We evaluate our NE extraction method on five classes, i.e., *star*, *film*, *TV play*, *song*, and *PC game*. The reason to select these classes is that they are among the most frequently searched in search engines.

### 4.1 Experimental Data

In the experiments, we mainly evaluate the proposed method on Chinese. However, we also test the effectiveness of the method on English (Section 5.3). We therefore prepare experimental data for both languages.

We run our model on approximately 9.7 billion Chinese web titles and 13 billion English web titles respectively. Chinese web titles were collected from high-quality webpages after spam filtering and pageranking while English web titles were taken from all of our crawled English webpages. Note that, although the English corpus is larger than the Chinese one, it is still noisy and more sparse, given the fact that there are much more English (56.6%) webpages than Chinese (4.5%) on the whole internet<sup>4</sup>. The English titles are lowercased in preprocessing.

### 4.2 Metrics

We evaluate the methods based on precision ( $P$ ), coverage ( $C$ ), as well as the volume ( $V$ ) of the extracted NEs. In particular, precision is defined as the percentage of correct NEs of a given class from the automatically extracted ones. Precision is manually evaluated, in which we randomly sample 100 NEs from each resulting NE set of a given class, and ask two annotators to independently annotate whether each extracted NE belongs to the target class. The samples with different annotations are then reviewed by both annotators to produce the final result.

<sup>4</sup>[http://en.wikipedia.org/wiki/Languages\\_used\\_on\\_the\\_Internet](http://en.wikipedia.org/wiki/Languages_used_on_the_Internet) world.

Recall is more difficult to assess. Inspired by (Etzioni et al., 2004), we evaluate coverage against benchmark NE repositories. More specifically, we select a popular website for each given category in the corresponding language. For example, we use IMDB as the benchmark NE repository for categories *star*, *film* and *TV play* in English. All of the websites for constructing benchmark data on both Chinese (CH) and English (EN) are summarized in Table 1.

	Class	Website	Vol
CH	star	yule.sohu.com/star	4,643
	film	mtime.com	25,457
	TV play	www.mtime.com	4,080
	song	music.baidu.com	126,127
	PC game	pc.pcgames.com.cn	7,711
EN	star	imdb.com/	545,853
	film	imdb.com/	160,188
	tv play	imdb.com/	19,823
	song	spotify.com/	171,270
	pc game	gamespot.com/pc	8,131

Table 1: Benchmark dataset

The benchmark NEs were extracted from the websites using handcrafted patterns. Post-processing is done for the Chinese data, including discarding films and TV plays scored by only one viewer and songs played no more than 10 times. These filtering clues are extracted from the websites along with the NEs. For the English data, NEs are limited to those beginning with English characters and consisting of only English characters and some specific symbols (‘. – ;, !#). All English NEs are lowercased. The last column of Table 1 shows the statistics of the benchmark data. Coverage is computed as the percentage of NEs in a given benchmark set covered by the automatically extracted NEs. Please note that those websites for constructing benchmark data are not used in url patterns in the following experiments.

### 4.3 Results on Chinese

To extract Chinese NEs for the 5 examined classes, we first select 200 seeding NEs for each class. These seeds are randomly sampled from the top-5000 hot NEs for each class from Baidu<sup>5</sup> query logs. The results are shown in Table 2.

<sup>5</sup>[www.baidu.com](http://www.baidu.com), the largest Chinese search engine in the

Category	P	C	$C_{hot}$	Vol
star	0.99	0.85	0.90	7,630
film	0.95	0.76	0.86	24,183
TV play	0.92	0.82	0.93	21,655
song	0.96	0.12	0.33	11,011
PC game	0.96	0.50	0.75	14,049
<b>average</b>	0.96	0.61	0.75	15,706

Table 2: NEM results on the Chinese corpus

	star	film	TV play	song	PC game
CH	0.52	0.53	0.86	0.11	0.79
EN	0.78	0.82	0.87	0.78	0.84

Table 3: Percentage of NEs out of benchmark dataset

As can be seen from the Table 2, the precision of the extracted NEs is pretty high, which exceeds 0.92 on all five classes. On the other hand, the coverage varies across different classes. Especially, the coverage on songs is very low, which is only 0.12. After observing the extraction patterns, we found that the low coverage of songs is mainly due to the complexity of the patterns. Specifically, the titles of music websites usually contain not only the song’s name, but also the singer’s name. For example, the title “青花瓷-周杰伦-在线试听mp3下载-酷我音乐(*Green Flower Porcelain - Jay Chou - online audition mp3 download - kuwo music*)” is from a music website “www.kuwo.cn”, in which “青花瓷(*Green Flower Porcelain*)” is the song’s name, while “周杰伦(*Jay Chou*)” is the singer’s name. The singer’s name may seriously influence the generality of the induced text patterns.

We further evaluate our method’s coverage of hot NEs. Here an NE is deemed hot if its daily search frequency is no less than 10 according to our query logs. The fourth column ( $C_{hot}$ ) of Table 2 depicts the results. We can see that the coverage of hot NEs is evidently higher than that of random NEs for all five categories. The volume of extracted NEs for each class is listed in the last column of Table 2. Furthermore, row 1 of Table 3 depicts the percentage of the extracted Chinese NEs that are out of the benchmark dataset, from which we can see that our method actually mines a lot of NEs that are not covered by the benchmark data. This demonstrates the importance of extracting NEs from multiple websites.

#### 4.4 Comparison Results

In this section, we compare our method with the method proposed by (Paşca, 2007b). Paşca’s method is guided by a small set of seed instances for each class. The method extracts NEs from user queries in 5 steps: (1) generating query patterns matching the seed instances, (2) identifying candidate NEs using the patterns, (3) representing each candidate NE with a vector of patterns extracting it, (4) representing each class with a vector of patterns extracting its seeds, (5) computing the similarity between the representing vectors of each candidate NE and the class, and ranking the candidate NEs according to the similarity. Extracting NEs from query logs is a promising direction since search queries reflect the netizen’s true requirements. In our experiments, we implement Paşca’s method using our query log data, which contains a total of 100 million Chinese queries from Baidu search engine. The seeds used here are the same as in our method.

	Class	P@500	P@5k	P@all	C
Paşca’s method	star	0.83	0.53	0.29	0.62
	film	0.95	0.79	0.73	0.10
	TV play	0.96	0.59	0.39	0.32
	song	0.98	0.92	0.86	0.12
	PC game	0.73	0.34	0.11	0.28
	<b>average</b>	0.89	0.63	0.48	0.29
Our method	star	0.98	0.98	0.99	0.85
	film	1.00	1.00	0.95	0.76
	TV play	0.99	1.00	0.92	0.82
	song	1.00	0.94	0.96	0.12
	PC game	1.00	0.97	0.96	0.50
	<b>average</b>	0.99	0.98	0.96	0.61

Table 4: Comparison with Paşca (2007b)’s method

We compare two methods based on precision at different numbers of extracted NEs, by annotating 100 NEs out of the first 500, 5k and all respectively, as well as coverage. The comparison results are shown in Table 4. We can find from the result that our method significantly outperforms Paşca’s in both precision and coverage (C). Especially, the precision of NEs extracted by Paşca’s method sharply decreases when lower ranked NEs are examined, whereas the quality of NEs extracted by our method seems quite stable.

## 5 Analysis

### 5.1 Analysis of Experiment Settings

This section analyzes the influence of the experimental settings. We first introduce the performance when using text pattern only, and then examine the contribution of the inter- and intra-class scoring in the MCL learning. Finally, we show how the performance varies with different number of iterations. Table 5 shows the P@500, P@5k

Class	P@500	P@5k	P@all	C
star	0.97	0.85	0.62	0.31
film	0.98	0.98	0.97	0.20
song	0.89	0.70	0.37	0.08
TV play	0.96	0.93	0.72	0.23
PC game	0.56	0.18	0.11	0.01

Table 5: Performance when using text pattern only

and P@all performance when only using text patterns. The precision seems relatively good but the coverage is generally low. The precision falls rapidly as the number of selected NEs grows except the category *film*. This table indicates that url patterns play an important role in our method, without which the quality of the extracted NEs cannot be guaranteed.

Table 6 shows the P@500 and P@5k performance of our method when we only use intra-class or inter-class scoring in MCL learning. We can find that there is a dramatic decrease in the performance in both settings, suggesting that both inter-class and intra-class scoring criterion are necessary to guarantee the accuracy of the extracted NEs, and they should be used together.

Table 7 shows the performance after 1, 3, and 5 iterations. The number of url patterns is also listed along with precision and coverage. As can be seen, the average precision only slightly drops from 0.97 to 0.96 after 5 iterations, whereas the average coverage increases significantly from 0.53 to 0.61. This is mainly because the extraction sources grow almost 3 times, from 314 url-patterns to 1129 for each category on average.

### 5.2 Error Analysis

We have analyzed the erroneous NEs extracted by our method. This paragraph analyzes errors regarding precision while the following paragraph describes errors about recall. It turns out that ambiguity is a main reason for the errors. We find

	Category	P@500	P@5k
Using only intra-class estimator in MCL	star	0.11	0.36
	film	0.17	0.18
	TV play	0.12	0.43
	song	0.09	0.12
	PC game	0.43	0.10
	<b>average</b>	0.18	0.24
Using only inter-class estimator in MCL	star	0.07	0.08
	film	0.97	0.97
	TV play	0.98	0.96
	song	0.25	0.35
	PC game	0.75	0.79
	<b>average</b>	0.60	0.63

Table 6: Performance when using only intra-class or inter-class scoring in MCL

it quite common that an NE belongs to more than one class. For example, a TV play might be adapted from a novel with the same name, a biographical film might be named after the protagonist, etc. Statistics reveal that in “www.mtime.com”, which is the benchmark data for extracting Chinese films and TV plays in our work, 12.8% of the TV plays have homonymic films in the same website, while the percentage is 14% in its English counterpart “www.imdb.com”. Our method suffers from the ambiguity problem since the homonymic NEs might yield url-text patterns belonging to other classes, and thereby bring in noisy NEs.

Besides, as pointed out in Section 4.3, title complexity is the main problem that hinders NE extraction. Particularly, some titles contain more than one NE, which makes it difficult to induce text-patterns for a certain class from these titles. Another reason leading to the mismatch of benchmark NEs is that some NEs have different forms in different sources. For instance, we extract the song “*New Years Project*”, but the correct form in the benchmark data is “*New Year’s Project*”.

### 5.3 Language Adaptation

Our method is language-independent. This section presents the evaluation in English. The English data is described in Section 4.1. Table 8 shows the performance of our method.

We can see from the table that the precision of the extracted English NEs is also high. Compared with Table 2 above, we can find that the coverage of the English NEs is lower than that on Chinese. However, the volume of the extracted NEs is al-



#Iteration	measure	star	film	TV play	song	PC game	average
1	Precision	0.98	0.98	0.99	0.98	0.92	0.97
	Coverage	0.86	0.55	0.65	0.12	0.47	0.53
	url-text patterns	296	437	387	243	207	314
3	Precision	0.98	0.96	0.95	0.97	0.91	0.95
	Coverage	0.89	0.75	0.66	0.12	0.53	0.59
	url-text patterns	745	2,041	1,525	324	659	1,059
5	Precision	0.99	0.95	0.92	0.96	0.96	0.96
	Coverage	0.85	0.76	0.82	0.12	0.50	0.61
	url-text patterns	791	2,101	1,593	325	835	1,129

Table 7: Performance for varying number of iterations

Category	P	C	Vol
star	0.98	0.65	1,589,002
film	0.92	0.40	352,152
TV play	0.89	0.59	71,273
song	0.95	0.31	240,335
PC game	0.97	0.80	29,166
<b>average</b>	0.94	0.55	456,386

Table 8: Performance on English corpora

most 30 times larger. This is unsurprising, since there are much more NEs written in English than in Chinese on the internet. Given that the English corpus used in our experiments is only 1.4 times larger than the Chinese corpus, we believe that data sparseness might be a major cause of the low coverage. Likewise, from the row 2 of Table 3, our method also acquires a large proportion of NEs in English which do not exist in the benchmark websites.

To have a better understanding of the coverage problem, we examined the cases not extracted with our method. As we have analyzed the problem of song, we randomly sampled 1000 NEs missed by our method for the other 3 classes with low coverage on the Chinese test dataset, i.e., *star*, *film*, and *TV play*. We then examined whether the missed cases contain a lot of hot NEs according to the following heuristics: (1) If a star has no picture on the imdb page, then it should not be deemed hot. Our statistics show that 97.3% missed stars have no pictures. (2) If a film’s duration is no longer than one hour and the number of viewers grading it on IMDB is less than 10, then the film should not be hot. 90.6% missed films are not hot according to this criterion. (3) Similar to film, if the number of reviewers grading a TV play is less than 10, then

it is not hot. 69.9% of the missed TV plays are not hot accordingly. On the whole, the above numbers suggest that the NEs not covered by our method are mostly unpopular ones, which may seldom be used in real applications.

## 6 Conclusion and Future Work

In this paper, we propose to extract NEs from web document titles using url-text hybrid patterns. A multiclass collaborative learning mechanism is introduced into the bootstrapping algorithm to better perform the quality control. We evaluate our method on five categories popular in real applications, in both Chinese and English. The results reveal that the precision and coverage (against benchmark data) of the extracted NEs are 0.96 / 0.61 in Chinese, and 0.94 / 0.55 in English. Detailed analysis demonstrates that the url-text hybrid patterns are superior to conventional text wrappers, and the multiclass collaborative learning mechanism is effective. Further comparison also shows that our method can significantly outperform a representative method that learns NEs from query logs.

Our future work will be carried out along two directions, i.e. improving the text-pattern induction approach and testing the method in more other languages.

## 7 Acknowledgment

This work is supported by National High-tech R&D Program of China (863 Program) under the grant number: 2011AA01A207. We give warm thanks to Prof. Jian-Yun Nie and other anonymous reviewers for their comments.

## References

- Lorenzo Blanco, Nilesh Dalvi, and Ashwin Machanavajjhala. 2011. Highly efficient algorithms for structural clustering of large websites. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 437–446, New York, NY, USA. ACM.
- William W. Cohen and Sunita Sarawagi. 2004. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 89–98, New York, NY, USA. ACM.
- Bhavana Bharat Dalvi, William W. Cohen, and Jamie Callan. 2012. Websets: extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 243–252, New York, NY, USA. ACM.
- Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artificial intelligence*, IJCAI'07, pages 2733–2739, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 100–110, New York, NY, USA. ACM.
- Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134, June.
- Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *ACL*, pages 407–415.
- Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1110–1118, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Marius Paşca. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 137–145, New York, NY, USA. ACM.
- Marius Paşca. 2007a. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 101–110, New York, NY, USA. ACM.
- Marius Paşca. 2007b. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 683–690, New York, NY, USA. ACM.
- Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, AAAI '99/IAAI '99, pages 474–479, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- Partha Pratim Talukdar, Thorsten Brants, Mark Liberman, and Fernando Pereira. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X 06, pages 141–148, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 214–221, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Richard C. Wang and William W. Cohen. 2008. Iterative set expansion of named entities using the web. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 1091–1096, Washington, DC, USA. IEEE Computer Society.
- Richard C. Wang and William W. Cohen. 2009. Automatic set instance extraction using the web. 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 - Volume 1*, ACL '09, pages 441–449, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Feature-Rich Segment-Based News Event Detection on Twitter

Yanxia Qin<sup>1</sup>   Yue Zhang<sup>2</sup>   Min Zhang<sup>3,1</sup>   Dequan Zheng<sup>1</sup>

<sup>1</sup> School of Computer Science and Technology,  
Harbin Institute of Technology, Harbin 150001, China

<sup>2</sup> Singapore University of Technology and Design, Singapore 138682

<sup>3</sup> School of Computer Science and Technology,  
SooChow University, Suzhou 215006, China

{yxqin, dqzheng}@mtlab.hit.edu.cn

yue\_zhang@sutd.edu.sg   zhangminmt@hotmail.com

## Abstract

Event detection on Twitter is an important and challenging research topic. On the one hand, Twitter provides first-hand information and fast broadcasting. On the other, challenges include short and noisy content, big volume data and fast-changing topics. Dominant approaches for Twitter event detection model events by clustering tweets, words or segments, while segments have been proven to be advantageous over both words and tweets in news event detection. We study segment-based news event detection, for which existing heuristic-based methods suffer from low recall. We propose feature-based event filtering to address this issue. Our filter incorporate a rich family of features that are empirically proven to be valuable. Experimental results show that our event detection system outperforms the state-of-the-art baseline with doubled recall and increased precision.

## 1 Introduction

We study news event detection from Twitter messages (tweets). Generally, tweets can be classified into three groups: 1) *news events*, or breaking news such as “Manchester united Vs Athletic in Jan. 1st”; 2) *hot topics* that spread among a large amount of Twitter users, such as horoscope topics (e.g. “You have recently experienced a phase of expansionism and it’s... More for Sagittarius”); and 3) *heterogeneous collections* or, meaningless non-event tweets, such as “Need buddy wanna chat”. Some previous work (Cataldi et al., 2010; Kasiviswanathan et al., 2011; Diao et al., 2012) regards both news events and hot topics as subjects of detection, while other work (Jackoway et al., 2011; Sakaki et al., 2010; Becker et al., 2012)

only detects news events. We are interested in the latter, for which most previous work detects only specific types of events. For example, Sakaki et al. (2010) detect earthquake events from Twitter. In this paper, we study event detection in general.

Compared with event detection in news texts, Twitter provides more opportunities and challenges. Yom-Tov and Diaz (2011) report that Twitter can broadcast news faster than traditional media, which provides an opportunity for event detection in Twitter. On the other hand, there are challenges in event detection from Twitter data: 1) tweets are too short and sometimes cannot carry enough information; 2) tweets contain many noisy words, which can be harmful for event detection and 3) the volume of Twitter data is very large, which makes event detection a big data problem.

The dominant approach for Twitter event detection is clustering. Similar tweets (Becker et al., 2011; Li et al., 2012c) or words (Platakis et al., 2009; Lee et al., 2012) are group into a cluster, before clusters are classified into either news events or non-events. A recent paper (Li et al., 2012a) showed that segments (i.e. ngrams; see Section 2) can be advantageous over both tweets and words for clustering. As segments have much smaller quantity than tweets and are more semantically meaningful than words, they are better units to be clustered. We take Li’s system (Twevent) as our baseline system.

Twevent apply a heuristic-based method (*newsworthiness*) to filter out hot topics and heterogeneous clusters from news events. *Newsworthiness* is calculated by similarity between edges in a cluster, and whether segments of the cluster frequently appear in Wikipedia. Both similarity of edges and Wikipedia are useful in filtering out heterogeneous collections from news events, while Wikipedia can also separate some news and topics. However, there are several problems with this approach: 1) *newsworthiness* cannot distinguish

news from some topics, includes horoscope topics (“sagittarius; approach; big trouble”) and topics such as “hitler; fox; megan fox; rip; megan; selena gomez”, which contain segments that can also frequently occur in Wikipedia; 2) as a single measure, *newsworthiness* is subject to a tradeoff between precision and recall, while a high precision can be obtained only with an extremely low recall (about 10%).

On the other hand, tweets contain useful information that can address the weakness of *newsworthiness*. For example the “Follow spree” topic, which refers to following-back activities by celebrities to their fans, can be recognized by the common hashtag suffix “followspree”. Another example is that news tweets are more likely to contain url links. We propose a classifier based method for event filtering and define a set of novel features that capture statistical, social and textual information from event clusters. Some of the features are useful in getting rid of heterogeneous collections while others may be useful for recognizing news events from hot topics.

We call our system Feature-Rich segment-based news Event Detection system on Twitter (FRED). Experimental results show that our system FRED outperforms the state-of-the-art event detection system Twevent by significantly increased precision and doubled recall.

## 2 Segment-Based Event Detection

In this section, we introduce the segment-based event detection method of Li et al. (2012a), which consists of three steps: tweet segmentation, bursty segment detection and segment clustering. Tweet Segmentation splits tweet into non-overlapping segments, which maybe unigrams or N-grams (2-5 grams). For a certain time window, segments that show a bursty frequency pattern are selected as bursty segments. Segment clustering groups bursty segments about same event into one cluster regarding them as one event.

### 2.1 Tweet Segmentation

Tweet segmentation can be regarded as a optimization problem to partition tweet with the use of Microsoft Web N-Gram service<sup>1</sup> and Wikipedia<sup>2</sup>.

<sup>1</sup><http://web-ngram.research.microsoft.com/info/>

<sup>2</sup><http://www.wikipedia.org/>

The objective function is defined as:

$$\arg \max_{s_1, \dots, s_m} C(d) = \sum_{i=1}^m C(s_i) \quad (1)$$

where  $d$  is a tweet from Tweet stream,  $\{s_1, \dots, s_m\}$  are the segments in tweet  $d$  and  $C$  is the function which measures the stickiness of a tweet or segment. In particular:

$$C(s) = L(s) \cdot e^{Q(s)} \cdot S(\text{scp}(s)) \quad (2)$$

where the length  $L(s)$  is defined in Eq 3, and a longer  $s$  makes it typically less sticky.  $Q(s)$  is the probability that  $s$  appears as an anchor text in Wikipedia articles; frequently-appearing anchor texts are more semantically meaningful.  $S(\cdot)$  is the sigmoid function.  $\text{scp}(s)$  is a cohesiveness measurement of segment  $s$  defined with symmetric conditional probability, as shown in Eq 4. Better combination of words when forming segments leads to higher cohesiveness value.

$$L(s) = \begin{cases} \frac{|s|-1}{|s|}, & \text{for } |s| > 1 \\ 1, & \text{for } |s| = 1 \end{cases} \quad (3)$$

$$\text{scp}(s) = \log \frac{Pr(s)^2}{\frac{1}{n-1} \sum_{i=1}^{n-1} Pr(w_1^i) Pr(w_{i+1}^n)} \quad (4)$$

In the above equations, a segment  $s$  can be written as  $\{w_1 \dots w_n\} (n > 1)$ , where  $Pr(\cdot)$  is the prior probability derived from the Microsoft Web N-gram service.

### 2.2 Bursty Segment Detection

From the large number of segments resulting from the last step, a small portion of bursty ones are selected for event clustering since segments with a burst frequency are more representative for a breaking news in the data stream. For convenience, we take a time window  $t$  as the time unit for bursty segment detection and segment clustering.  $N_t$  refers to the number of tweets within the time window  $t$ , and  $f_{s,t}$  represents the number of tweets that contain segment  $s$  within  $t$ . If  $f_{s,t} > E[s|t]$ , then a segment  $s$  is a **bursty segment**.  $E[s|t]$  is the expected number of tweets that contain  $s$  within  $t$ . As  $N_t$  is sufficiently large, the Gaussian distribution is used to model the probability of  $f_{s,t}$ .

$$P(f_{s,t}) \sim N(N_t p_s, N_t p_s (1 - p_s)) \quad (5)$$

where  $p_s$  is expected probability of tweets contain  $s$ , calculated as:

$$p_s = \frac{1}{L} \sum_{t=1}^L \frac{f_{s,t}}{N_t} \quad (6)$$

$L$  is number of time windows containing  $s$ .

$E[s|t] = N_t p_s$ . Even after filtering out non-bursty segments, a large amount of bursty segments remain. **Bursty weight**  $w_b(s, t)$  is assigned to each bursty segments and the top  $K$  bursty segments are chosen for further processing.  $K$  is set to  $\sqrt{N_t}$  in Twevent.

$$w_b(s, t) = P_b(s, t) \log(u_{s,t}) \quad (7)$$

$P_b(s, t)$  is the bursty probability and  $u_{s,t}$  means the **user frequency** of  $s$  helping to filter out some noisy segments, as the more users talk about the segment  $s$ , the more popular and meaningful it is.  $u_{s,t}$  is calculated as the number of users who post tweets containing  $s$  within  $t$ .

$$P_b(s, t) = S(10 \times \frac{f_{s,t} - (E[s|t] + \sigma[s|t])}{\sigma[s|t]}) \quad (8)$$

$\sigma[s|t] = \sqrt{N_t p_s (1 - p_s)}$  is the standard deviation of Gaussian distribution in Eq 5.

### 2.3 Segment Clustering

k-Nearest Neighbor graph (kNNgraph) clustering method, is applied to group bursty segments into clusters. The kNNgraph clustering method takes a complete graph of bursty segments with edges representing similarity between segments as input and output event clusters. It groups two segments into same cluster only when they are in each other's k-nearest neighbors.  $k$  is a key parameter to control the size of clusters. We choose value for  $k$  in Section 4. The output of kNNgraph clustering is an event cluster set corresponding to the time window  $t$ , denoted as  $G_{set}(t)$ . All  $G_{set}(t)$  sets are gathered to a whole event cluster set  $G_{set}$ .  $G_{set}$  is manually labeled for further use, introduced in Section 4.2.

Temporal features and text similarity are incorporated when calculating similarity between two segments  $s_1, s_2$ .

$$sim_t(s_1, s_2) = \sum_{m=1}^M w_t(s_1, m) w_t(s_2, m) sim(T_1, T_2) \quad (9)$$

$\langle t_1 \dots t_M \rangle$  are  $M$  sub time windows of the time window  $t$ . Frequency of segment  $s$  in the sub time window  $t_m$  is denoted as  $f_t(s, m)$ .  $w_t(s, m)$  is the frequency weight of  $s$  in  $t_m$ , which serves as a temporal feature and is shown in Eq 10.  $T_i$  denotes a set of tweets containing  $s_i$  within  $t_m$ .  $sim(T_1, T_2)$  measures text similarity between the two sets of tweets  $T_1, T_2$ . Tweets in  $T_i$  are concatenated as a pseudo document, and cosine similarity is applied for calculating distance. Pseudo documents are represented by the Vector Space Model, weighted by TF-IDF. TF value is the number of tweets containing word  $w$  within the sub time window  $t_m$  and DF value is the number of tweets containing  $w$  in the whole twitter corpus.

$$w_t(s, m) = \frac{f_t(s, m)}{\sum_{m'=1}^M f_t(s, m')} \quad (10)$$

## 3 Feature-Rich Event Filter

The clusters in the kNNgraph clustering result  $G_{set}$  contain news events, hot topics and heterogeneous clusters, corresponding to the three types of tweets mentioned in the Introduction. Our goal, which is to recognize news events from other two types of event clusters, is a challenging task because hot topics and news events can both have bursty frequency and share similar characteristics.

### 3.1 Event Filter in Twevent

Twevent utilizes a heuristic-based method for event filtering using information from Wikipedia. A heuristic equation, *newsworthiness*, is used to determine whether an event cluster is a news event or not, whereas all clusters with a high *newsworthiness* score is news events. The *newsworthiness*  $\mu(e)$  of an event cluster  $e$  containing segment set  $S_e$  and edge set  $G_e$  is calculated as follows.

$$\mu(e) = \frac{\sum_{s \in S_e} \mu(s)}{|S_e|} \cdot \frac{\sum_{g \in G_e} sim(g)}{|S_e|} \quad (11)$$

where  $\mu(s)$  of segment  $s$  is calculated as:

$$\mu(s) = \max_{l \in s} e^{Q(l)} - 1 \quad (12)$$

$l$  is sub-phrase of  $s$  and  $Q(l)$  is the probability that  $l$  appears as anchor text in Wikipedia articles.

An event cluster  $e$  is taken as a news event only if it satisfies the condition that  $\mu_{max}/\mu(e) < \tau$ , where  $\tau$  is a threshold for *newsworthiness*,  $\mu_{max}$

is the maximum of  $\mu(e)$  in time window  $t$ . Lower  $\tau$  value leads to high precision and low recall, which is the limitation of *newsworthiness*. Rich information from tweets and clusters themselves can be useful in alleviating this problem.

### 3.2 Event Filter in FRED

In order to incorporate rich features, we take event filtering as a binary classification problem, where class ‘T’ means true news event class includes news events and class ‘F’ represents false news event class containing hot topics and heterogeneous clusters. In our filter, event clusters in  $Gset$  are represented with a set of cluster-level features, and classified into T or F by a SVM<sup>3</sup> classifier. All clusters in class T, represented as  $Eset$ , form the final news event result. Features used to represent event clusters are shown in Section 3.3.

### 3.3 Features

We collect three types of features for the filter, representing statistical, social and textual information related of event clusters, respectively. Some of the features are designed to filter out heterogeneous clusters, while others to distinguish news events from hot topics. Given an event cluster  $e$  and the corresponding time window  $t$  (from which  $e$  is extracted), we have the following information: 1)  $Gset(t)$ , a sub set of  $Gset$  corresponding to  $t$ . 2)  $S_e$ , the set of segments in  $e$  and  $G_e$ , the set of edges in  $e$ . 3)  $T(e)$ , which consists of tweets that are related to  $e$  containing at least one segment of  $S_e$  and being posted in  $t$ . 4)  $relU(e)$ , which represents users who posted the tweets in  $relT(e)$ , and  $U_t$ , which denotes the number of users who published tweets within  $t$ .

#### Statistical Features

For statistical features, we collect direct statistical information from event clusters, such as how many segments and edges it contains, the density of the event graph and so on.

- *seg*, which refers to the segment number of  $e$ , calculated as  $|S_e| / \max_{e' \in Gset(t)} (S_{e'})$ . News events and hot topics contain more segments than heterogeneous clusters generally.
- *edge*, which refers to number of the edges of  $e$ , defined as  $G_e / \max_{e' \in Gset(t)} (G_{e'})$ . Similar

with segment number, heterogeneous clusters usually have less edges than news and topics.

- *wiki*, the average of *newsworthiness* for all segments in  $S_e$ . A higher *wiki* value indicates that the event cluster contains more meaningful and important segments. *wiki* is able to distinguish news events from hot topics and heterogeneous clusters, as shown Li et al. (2012a).
- *dup* is designed to filter out some specific heterogeneous clusters that contains words sharing the same lemma. For example the event cluster  $e7$  in Table 3, which words sharing lemma “feel”. *dup* can be obtained by stemming all unigrams appeared in  $S_e$  and calculating the number of duplicated stemmed unigrams out of all stemmed unigrams.
- *sim*, which refers to average similarity of all edges in  $G_e$ . A bigger *sim* means that the event cluster is more dense, or sticky.
- *df*, which refers to the number of tweets related to  $e$  out of all tweets published in  $t$ , namely  $|relT(e)|/N_t$ . *df* could help to eliminate heterogeneous clusters, which are published by less users in less tweets.
- *udf*, which refers to  $|relU(e)|/U_t$ . The influence of *udf* is similar with *df*.

#### Social Features

Tweets in  $relT(e)$  contain rich Twitter-specific social information, which may reveal the difference between news events and hot topics. For example, the more mentions (@username) exist in  $relT(e)$ , the more likely  $e$  is a topic.

- *rt*, which represent how many tweets in  $relT(e)$  are retweeted. *Retweet* is a forwarding action on a tweet published by other users indicating an interest to the tweet. A retweeted tweet is denoted by a prefix of “RT @username”. *Retweet* functions as a means of sharing and spreading without commenting to show user’s opinion. A news event may have a larger fraction of retweeted tweets than others as users want to spread the news.
- *men*, which refers to the normalized number of tweets containing *mention* (e.g., @username) in  $relT(e)$  specifying one target

<sup>3</sup>We use LibSVM <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> for the experiment.

receiver of tweets (e.g., “@justinbieber”). *Mention* actions occur more frequently in hot topics than in news events, as users prefer showing their opinion about this topic rather than just spreading it.

- *rep*, which refers to the normalized number of reply tweets in  $relT(e)$ . *Reply* means commenting, and a reply tweet is started with a mention. Similar with *mention*, *reply* has strong indication of conversation, and are more related to topics than news.
- *url*, which refers to the normalized number of tweets containing url link in  $relT(e)$ . *Url* shows extra information for tweet. News events contains more information than a topic, which may not be fully expressed in a short tweet, and hence url links are likely used to refer to the original article.
- *tag*, which refers to the normalized number of tweets containing *hashtag* in  $relT(e)$ . A *hashtag* (#gamecocks) is a short description of what’s happening. Generally a popular hashtag indicates a hot topic or an event (e.g., “The game got a little exciting today but we got the win! #gamecocks”).
- *pst*, which measures how many tweets contain words in past tense in  $relT(e)$ , normalized by  $|relT(e)|$ . News events are more likely to be described formally and with more words in past tense.

### Textual Features

Besides above groups of features, text information embedded in hashtag content are another valuable source of information. News events will more likely have common hashtags. For example, many tweets about “National Football League” games have a common hashtag “#NFL”. Twitter topic can have common prefixes or suffixes of hashtag. For example the “Follow spree” topic, which is mentioned earlier, may have a common hashtag suffix “followspree”.

- *fTag*, which represents how many hashtags appear in  $relT(e)$  are frequent hashtags. We extract a frequent hashtag list from whole Twitter data set by taking the top 2000 most frequently used hashtags.
- *psfx*. To obtain frequent hashtag prefixes/suffixes, we first filter out prefixes/suffixes

of all hashtags in the data that satisfy at least one of the following conditions: 1) less than 3 characters, 2) composed by repeating one character, 3) frequency lower than 200. After arranging the prefixes in alphabetical order, we keep only the longest prefixes for the same prefix pattern. Prefixes are ranked by frequency, and the top 2000 are taken as frequent hashtag prefixes. Similarly, we could extract 2000 most frequent hashtag suffixes. *psfx* and *sfx* are used to indicate how many hashtags tweets in  $relT(e)$  contain frequent prefixes or suffixes respectively. *psfx* is the combination of *psfx* and *sfx* by multiplying them.

## 4 Experiments

### 4.1 Data

The Twitter data we use were crawled from Twitter timeline, which is the real-time tweet stream containing all tweets published by Twitter users from January 1st to January 15th, 2013. After removing stops words, filtering out non-English tweets and null content tweets, the data set contains 31,097,528 tweets published by 16,331,133 users with 382,475 words.

Wikipedia data is used as an extra resource in the tweet segmentation tasks (Section 2.1) and event filtering (Section 3). We use the Wikipedia dump data<sup>4</sup> of February 4th, 2013. It includes 13,167,739 pages and 10,507,127 anchor entities that have 5 words length limit. These anchor entities’ anchor probability, i.e. the number of pages that entity  $e$  appears as anchor text divided by the number of pages containing entity  $e$ , are calculated at the very beginning.

### 4.2 Settings

We reproduced Twevent as our baseline system. Parameter  $\tau$  in Twevent and *gamma* in FRED are tuned for best performance on a development set, which consists all event clusters on Jan. 2nd and 5th. Time window  $t$  is set to be a day and  $M$  (in Eq. 9) is 12.  $k$  in kNNgraph clustering method is set to be 5, as a tradeoff of the number of event clusters and average number of segments in clusters.  $\tau$  in Twevent is tuned and set to be 2 and *gamma* in LibSVM of FRED is 5. 10-fold cross validation is

<sup>4</sup>[http://burnbit.com/download/235406/enwiki\\_20130204\\_pages\\_articles\\_xml.bz2](http://burnbit.com/download/235406/enwiki_20130204_pages_articles_xml.bz2)

ExpID	FeatureSet	Precision	Recall	F1	Diff
0	All	83.64%	22.89%	35.94%	-
1	All- $\{seg\}$	82.73%	22.64%	35.55%	-0.39%
2	All- $\{edge\}$	82.35%	22.64%	35.51%	-0.43%
3	All- $\{df\}$	83.26%	22.89%	35.9%	-0.04%
4	All- $\{udf\}$	83.26%	22.89%	35.9%	-0.04%
5	All- $\{wiki\}$	78.57%	17.79%	29.01%	-6.93%
6	All- $\{dup\}$	82.88%	22.89%	35.87%	-0.07%
7	All- $\{sim\}$	77.78%	17.41%	28.46%	-7.48%
8	All- $\{rt\}$	82.51%	22.89%	35.83%	-0.11%
9	All- $\{men\}$	83.33%	22.39%	35.29%	-0.65%
10	All- $\{rep\}$	81.28%	22.14%	34.8%	-1.14%
11	All- $\{url\}$	82.38%	21.52%	34.12%	-1.82%
12	All- $\{tag\}$	82.35%	20.9%	33.33%	-2.61%
13	All- $\{pst\}$	83.78%	23.13%	36.26%	+0.32%
14	All- $\{ftg\}$	81.9%	22.51%	35.32%	-0.62%
15	All- $\{psfx\}$	83.56%	22.76%	35.78%	-0.16%

Table 1: Experimental Results Using Different Features.

utilized to get system-generated class labels for all event clusters.

We built a standard gold set for FRED after labeling the event cluster set  $Gset$ , which is the output of the segment-based event detection (Section 2). The labeling method is shown as follows. Given an event cluster  $e$ , the segments in  $e$  and the corresponding time window  $t$ , we use the segments and  $t$  to determine whether  $e$  is related to a news. Google and Twitter search are used to assist manual annotations of events. As a result, 4249 event clusters in  $Gset$  were manually labeled into 804 news events and 3445 non-events. Note that some news events in  $Gset$  may be sub events of one event. For example “The Golden Globe Awards ceremony 2013” happened in January 13th are detected more than once, as people talked about winners for different awards. We have not merged these sub events in this paper, which will be considered for future work.

With the event cluster set  $Gset$ , we use the precision, recall and F1-measure to evaluate the performances of FRED and Twevent, where precision is defined the fraction of news events in system-generated ‘T’ class event clusters ( $Eset$  for FRED), and recall measures how many manually labeled news events are detected out of all news events in  $Gset$ . F1 measure is calculated for an overall evaluation. Note that given our annotations, which is much larger than that of Li (Li et al., 2012a), we can give a better estimation of recall, which Li et al. were not able to report in detail (they used the number of detected news as recall, which did not reveal the real recall notion).

### 4.3 Experimental Results and Analysis

As we show some statistical results of tweet segmentation (Section 2.1), we obtained 1,604,129 distinct segments with 22.3% unigrams, 72% 2-grams and 5.7% 3-5 grams.

#### Effectiveness of Features

In Table 1 we show the results of feature ablation test. ExpID is the experiment id. FeatureSet is the features we used for current experiment. All means all statistical, social and textual features. Diff means the difference between F1 in current experiment and experiment 0, and a smaller Diff indicates that the feature is more valuable.

The experimental results show that nearly all features contribute to event filter on either precision or recall. Features can be partitioned into three groups according to their impact on precision and recall: 1) features that are useful only for precision include  $df$ ,  $udf$ ,  $dup$ ,  $rt$ . 2) features that are useful for both precision and recall includes rest of features such as  $wiki$ ,  $sim$ ,  $url$  etc. 3) feature  $pst$  is slightly harmful for precision and recall.

The most valuable features to our system are  $wiki$ ,  $sim$ ,  $rep$ ,  $url$  and  $tag$ .  $wiki$  is extra resource obtained from Wikipedia, and contributes to valuable segments in event clusters.  $sim$  indicates denser event cluster with stronger connections between segments, while replied tweet number, url number and hashtag number are social features embedded in tweets related to event clusters. Results show there are bigger differences in these features between news events and others clusters when compared to other social features.

#### The Performance of FRED



System	#Evt	P	R	F1
Twevent <sub>u</sub>	114	68.42%	9.7%	16.99%
Twevent	107	75.70%	10.07%	17.78%
FRED	146	83.64%	22.89%	35.94%

Table 2: Experimental Results.

The experimental results of FRED and baseline systems are presented in Table 2. Twevent<sub>u</sub> is a variant of Twevent, which uses unigrams (words) instead of segments in the event detection. #Evt is the number of news events.

The experimental result of Twevent (precision 75.7%) is lower than that reported by Li et al. (2012a) (precision 86.1%). It is likely to be caused by 1) different Twitter data, Li use Singapore Twitter data containing 4.3 million tweets in one month while ours is global Twitter data of 31.1 million tweets in half a month; 2) horoscope topics are very popular in our data, which cannot be filtered out by Twevent. Because horoscope topics greatly influence the performance of Twevent, we performed a manual filtering to them for a better result. Twevent without the extra process yields 125 event clusters with a low precision of 64.8%. No extra filtering process was necessary for FRED.

The results in Table 2 show that, 1) segments are better than words for news event detection as Twevent outperforms Twevent<sub>u</sub>, which brings in more heterogeneous collections; 2) our system FRED performs better than Twevent with significantly increased precision and doubled recall, which proves that feature-rich event filter could alleviate the low recall problem in Twevent.

### Analysis

We show some example event clusters in Table 3. Lm refers to manually annotated class label. Lt and Lf refer to class labels generated by Twevent and FRED, respectively. The labeling results in Table 3 show that Twevent and FRED made different types of mistakes. As mentioned earlier, Twevent (without manual filtering) always fails to distinguish horoscope topics, while FRED can. From e2-e3 and e4-e5, we can also see that Twevent’s labeling result changes for same news events while FRED gives consistent labels. Note that one important difference between FRED and Twevent is that the former uses some supervision. Preliminary experiments show that unsupervised

clustering such as k-means clustering cannot effectively bring the benefits of rich-features.

Football and basketball games, which appear almost everyday, take a large fraction of news events. Events such as the 27th Golden Globes Award ceremony hosted on January 13th, show bursty frequency patterns from late January 13th to 14th. Topics such as horoscope topics are popular everyday. At least from our data, the most popular hobbies of the globe seem to be football games.

Among all events, concert news or gossips about celebrities such as “Justin Bieber” and “Taylor Swift” draw much more and much longer attention. For example, e4 in Table 3, which is a news event related to Justin Bieber, continues to appear as news in many days very longer than e1 (a news related to song). New episode of TV programs and TV series such as “Big Brother” and “Pretty Little Liars” are also popular news events.

## 5 Related Work

Document-pivot clustering methods are frequently used in event detection on social media, in which short messages are regarded as documents (Becker et al., 2011; Li et al., 2012c). Becker et al. (2011) represent text content of a tweet as a TF-IDF weight vector and apply an incremental clustering algorithm to group similar tweets with one cluster regarded as an event. In the following event classification phase, temporal, social, topical and Twitter-centric features are used to represent each cluster and clusters are determined whether they are event-related or topic-related or non-event.

As social media data is on an extremely big scale, document-pivot clustering methods are ineffective as they are time- and memory-consuming. In contrast, in feature-pivot clustering methods, only features (words) that show a burst frequency pattern in a time window are extracted and then clustered into groups to get events. In addition to improving clustering efficiency, detecting bursty features also plays an important role for feature selection as social media messages are very noisy.

In most feature-pivot clustering methods, events are represented as a few representative words showing what happened, which may cause events to be difficult to understand (Li et al., 2012a; Platakis et al., 2009; Lee et al., 2012; Fung et al., 2005). Li et al. (2012a) adopted tweet segmentation in their event detection system Tweven-

ID	Lm	Lt	Lf	Time	Segments	Detail
e1	T	T	T	15th	golden disk awards; 27th; cr; kris; preview	Golden Disk Awards
e2	T	T	F	4th	lead; fans; check; vote; favorite music; peoples choice	peoples choice voting
e3	T	F	F	5th	lead; fans; check; vote; favorite music; peoples choice	related to e4
e4	T	T	T	2nd	paparazzi; chaos; accident; dangerous; fools; princess di	photographer died when chasing justin bieber
e5	T	F	T	3rd	paparazzi; town; sm; went	related to e6
e6	F	T	F	12th	venus; amorous; squares edgy	horoscope topic
e7	F	F	F	7th	feel; feel bad; feel i'm; feel sick	heterogeneous collection

Table 3: Example Events.

t. Tweet segmentation is firstly proposed by Li et al. (2012b) for an named entity recognition system on Twitter. They claim that segments are much more meaningful and easier to read than words. Twevent is the most related work to this paper. We adopt tweet segmentation, and segment tweets into non-overlapping segments that are regarded as bursty feature candidates, and utilize a feature-pivot clustering method to group bursty segments into clusters as events. The difference between this paper and Twevent is that they use a simple measurement (*newsworthiness*) to filter out meaningless twitter topics from events, while we propose a classifier based filter to distinguish news events and twitter topics. The advantage of our system is that it supports the definition of rich features, some of which are helpful to eliminate heterogeneous clusters and others can distinguish news events and hot topics. We will explore their functions in this paper.

In addition to the above group of work, which represents events with a few messages or features showing the topic information, some researchers try to extract structured information for events. Given a set of seed events, Benson et al. (2011) use a factor graph to extract artist and venue information of a concert event. Popescu et al. (2011) extract main entities, actions and audience opinions.

Data from social medias like Twitter are very sparse in presenting thousands of events, while some researchers mainly focus on specific types of events. Sakaki et al. (2010) detected disaster events like earthquakes and typhoons from Twitter. Pohl et al. (2012) tried to detect sub-event to assist disaster management with Flickr and YouTube data. Agarwal et al. (2012) analyzed tweets containing specific keywords and report Fire-in Factory and Labor-Strike events. They

have fixed query words and search for related messages from social media websites for data. The query words are challenges to define as they are vital to the quality of dataset, which will greatly influence the results. Becker et al. (2012) tried to generate queries for a planned event to relax the limitation. Our work mainly focus on news event detection problem on Twitter.

Rich features have been used in other tasks in NLP, such as POS-tagging (Toutanova et al., 2003), parsing (Zhang and Nivre, 2011) and machine translation (Chiang et al., 2009). Our work is in line with these.

## 6 Conclusion

We proposed a feature-rich classifier to recognize news events for segment based event detection, defining novel statistical, social and textual features for the filter. Experiments showed the effectiveness of the method, and in particular some features such as the number of urls and hashtags. The feature-rich event filter led to significantly higher precision and doubled recall when compared to the state-of-the-art baseline system. In our experiments we observed that a news event can be detected more than once in one time window, which each appearance representing one aspects of the event. Building these sub-events into a hierarchy will be explored in the future.

## Acknowledgments

Yanxia Qin is supported by SRG-SUTD2012038 from Singapore University of Technology and Design and the National Natural Science Foundation of China (No. 61073130) from Harbin Institute of Technology. Yue Zhang is fully supported by SRG-SUTD2012038.

## References

- Puneet Agarwal, Rajgopal Vaithiyathan, Saurabh Sharma, and Gautam Shroff. 2012. Catching the long-tail: Extracting local news events from twitter. In *ICWSM*.
- H. Becker, M. Naaman, and L. Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*.
- Hila Becker, Dan Iter, Mor Naaman, and Luis Gravano. 2012. Identifying content for planned events across social media sites. In *Proceedings of WSDM*, pages 533–542, Seattle, Washington, USA.
- Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *Proceedings of ACL-HLT*, pages 389–398, Portland, Oregon.
- Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. 2010. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of KDD*, pages 4:1–4:10, Washington, D.C.
- David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of NAACL*, pages 218–226, Boulder, Colorado.
- Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. 2012. Finding bursty topics from microblogs. In *Proceedings of ACL*, pages 536–544, Jeju Island, Korea.
- Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S. Yu, and Hongjun Lu. 2005. Parameter free bursty events detection in text streams. In *Proceedings of VLDB*, pages 181–192, Trondheim, Norway.
- Alan Jackoway, Hanan Samet, and Jagan Sankaranarayanan. 2011. Identification of live news events using twitter. In *Proceedings of LBSN*, pages 25–32, Chicago, Illinois.
- Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. 2011. Emerging topic detection using dictionary learning. In *Proceedings of CIKM*, pages 745–754, Glasgow, Scotland, UK.
- Sungjun Lee, Sangjin Lee, Kwanho Kim, and Jonghun Park. 2012. Bursty event detection from text streams for disaster management. In *Proceedings of WWW Companion*, pages 679–682, Lyon, France.
- Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012a. Twevent: segment-based event detection from tweets. In *Proceedings of CIKM*, pages 155–164, Maui, Hawaii, USA.
- Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012b. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of SIGIR*, pages 721–730, Portland, Oregon, USA.
- Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. 2012c. Tedas: A twitter-based event detection and analysis system. In *Proceedings of ICDE*, pages 1273–1276, Washington, DC, USA. IEEE Computer Society.
- Manolis Platakis, Dimitrios Kotsakos, and Dimitrios Gunopulos. 2009. Searching for events in the blogosphere. In *Proceedings of WWW*, pages 1225–1226, Madrid, Spain.
- Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2012. Automatic sub-event detection in emergency management using social media. In *Proceedings of WWW Companion*, pages 683–686, Lyon, France.
- Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Paranjpe. 2011. Extracting events and event descriptions from twitter. In *Proceedings of WWW*, pages 105–106, Hyderabad, India.
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of WWW*, pages 851–860, Raleigh, North Carolina, USA.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL*, pages 173–180, Edmonton, Canada.
- Elad Yom-Tov and Fernando Diaz. 2011. Location and timeliness of information sources during news events. In *Proceedings of SIGIR*, pages 1105–1106, Beijing, China.
- Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the ACL-HLT*, pages 188–193, Portland, Oregon.

# Building Chinese Event Type Paradigm Based on Trigger Clustering

Xiao Ding, Bing Qin, Ting Liu\*

Research Center for Social Computing and Information Retrieval

Harbin Institute of Technology, China

{xding, qinb, tliu}@ir.hit.edu.cn

## Abstract

Traditional Event Extraction mainly focuses on event type identification and event participants extraction based on pre-specified event type annotations. However, different domains have different event type paradigms. When transferring to a new domain, we have to build a new event type paradigm. It is a costly task to discover and annotate event types manually. To address this problem, this paper proposes a novel approach of building an event type paradigm by clustering event triggers. Based on the trigger clusters, the event type paradigm can be built automatically. Experimental results on three different corpora – ACE (small, homogeneous, open corpus), Financial News and Musical News (large scale, specific domain, web corpus) indicate that our method can effectively build an event type paradigm and can be easily adapted to new domains.

## 1 Introduction

Event extraction techniques have been widely used in several different specific domains, such as musical reports (Ding et al., 2011), financial analysis (Lee et al., 2003), biomedical investigation (Yakushiji et al., 2001) and legal documents (Schilder et al., 2007). Traditional event extraction systems achieved excellent performance in some important information extraction benchmarks, such as MUC (Message Understanding Conference, Chinechor et al., 1994) and ACE (Automatic Content Extraction). However, most of these methods require pre-specified event types as their prior knowledge. For example, ACE defines an event as *a specific occurrence involving participants*, and it annotates 8 types and 33 subtypes of events (LDC, 2005). However, building an event type paradigm in this way not only requires massive human effort but also tends to be very data dependent. As a result, it

may prevent the event extraction from being widely applicable. Since event types among domains are different, the event type paradigm of ACE, which does not define music related events, is useless for the music domain event extraction. So we have to build a totally different event type paradigm for the music domain from scratch.

Recently, some researchers have been aware of the limitations of only considering pre-defined paradigm as well. In the same vein, some studies work on the problem of relation extraction (Chambers and Jurasky, 2011 and 2009; Poon and Domingos, 2009 and 2008; Yates and Etzioni, 2009). Rosenfeld and Feldman (2006) built a high-performance unsupervised relation extraction system without target relations in advance. Hasegawa et al. (2004) discovered relations among named entities from large corpora by clustering pairs of named entities. However, most of the above work focuses on relation extraction rather than event extraction.

In contrast to the well-studied problem of relation extraction, only a few works focused on event extraction. For example, Li (2010) proposed a domain-independent novel event discovery approach. They exploited a cross-lingual clustering algorithm based on sentence-aligned bilingual parallel texts to discover event trigger clusters. Their motivation is to discover novel events for a new domain rather than build a new event type paradigm from scratch. So it takes domain specific event triggers as the input. However, it is also a costly task to annotate triggers for new domains.

To address above issues, this paper proposes a series of novel algorithms to automatically build event type paradigm. The proposed approach is based on the definition of event trigger: *the word that most clearly expresses an event's occurrence*, and our key observations: *triggers are the most important lexical units to represent events. A set of triggers with similar meaning or usage represents the same event type. Event types can*

---

\* Email correspondence

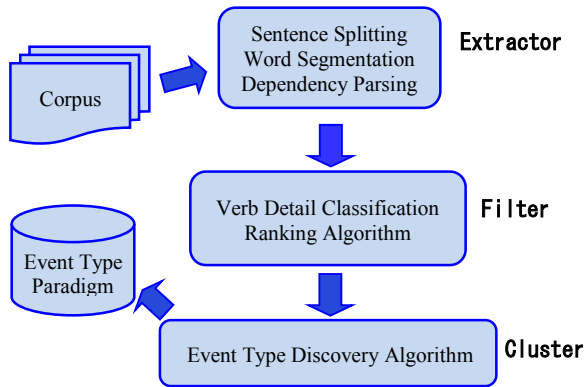


Figure 1. Architecture of the proposed system

be discovered based on trigger clustering. Our approach involves three steps: 1) we introduce a trigger extraction algorithm based on the dependency syntactic structure; 2) a trigger filter is then constructed to remove some noisy candidate triggers; 3) we develop an event type discovery algorithm based on our proposed trigger clustering methods. The clustered event types are used to construct an event type paradigm.

Experimental results show that our approach not only achieve significantly better performances than the baseline method, but also are more stable across different corpora. On the ACE, Financial News and Musical News corpus, the average accuracy is 73%. It shows that trigger clustering based method is effective on building an event type paradigm which is the premise of event extraction. We extract 33 event types for the ACE corpus, nine event types for the Financial News corpus and seven event types for the Musical News corpus.

Our contributions are as follows.

1. In this paper, we put forward the problem of event type paradigm building, and develop a novel framework as the solution.
2. This paper exploits a series of novel algorithms for automatically discovering and clustering domain independent event types.

The remainder of this paper is organized as follows. Section 2 presents our approach for event type paradigm building. Section 3 evaluates the proposed method. The related work on event extraction is discussed in Section 4, and we conclude the paper in Section 5.

## 2 Approach Overview

Since event trigger is the word that most clearly expresses an event’s occurrence, the key idea of this paper is to automatically construct an event type paradigm by clustering event triggers. For example, in the ACE corpus, a set of event triggers {倒闭, 闭门, 关闭, 停业, 解散} ({bankrupt,

Algorithm 1: TE algorithm	
<b>Input:</b>	Raw corpus D
<b>Output:</b>	Candidate triggers
1:	<b>ForEach</b> document d in raw corpus D <b>Do</b>
2:	d ← Paragraph Splitting
3:	d ← Sentence Splitting
4:	<b>ForEach</b> sentence s in document d <b>Do</b>
5:	s ← Word Segmentation
6:	s ← Chinese Dependency Parsing
7:	s ← Identify subject-predicate relation ( <i>SBV</i> ) pair ( <i>V<sub>SBV</sub></i> , <i>Sub</i> ) and verb-object relation ( <i>VOB</i> ) pair ( <i>V<sub>VOB</sub></i> , <i>Obj</i> )
8:	<b>If</b> $V_{SBV} = V_{VOB} = V_i$ , <b>Then</b>
9:	Extract $V_i$ as candidate trigger
10:	<b>End For</b>
11:	<b>End For</b>

Figure 2. The algorithm for trigger extraction

shut down, close, close down, dismiss}) represents the sense of the event type “Business/End-Org”. As shown in Figure 1, our system has three main components: trigger extractor, trigger filter and trigger cluster. The input of the system is a raw corpus, such as the ACE corpus, the Financial News corpus and the Musical News corpus, and the output is the event type paradigms which are shown in Table 2, Table 3 and Table 4.

### 2.1 Trigger Extractor

An event trigger is the center of an event, which is an important feature for recognizing the event type. Kiyoshi Sudo (2003) summarized three classical models for representing events. All of these three models rely on the syntactic tree structure and the trigger is specified as a predicate in this structure. In order to accurately extract event triggers, we employ the predicate-argument model (Yangarder et al., 2000) which is based on a direct syntactic relation between a predicate and its arguments. We extract the syntactic relation for predicate-argument model by means of the HIT (Harbin Institute of Technology) Dependency Parser (Che et al., 2009). Based on the predicate-argument model, we propose a trigger extraction algorithm (TE). The details are shown in Figure 2.

Take the following sentence as an example:

毛泽东 1893 年 出生 于 湖南湘潭。  
<sub>1 2 3 4 5</sub>  
→ Mao Zedong was born in Xiangtan, Hunan  
<sub>1 2 3 4 5</sub>  
Province in 1893.  
<sub>6 7</sub>

The HIT Chinese Dependency Parser dependencies are:

*SBV* (出生-3, 毛泽东-1)  
→ (born-3, Mao Zedong-1)  
*VOB* (出生-3, 湖南湘潭-5)  
→ (born-3, Xiangtan, Hunan Province-5)

ADV (出生-3, 1893 年-2)

→ (born-3, 1893-7)

POB (湖南湘潭-5, 于-4)

→ (Hunan Province-5, in-4)

where each atomic formula represents a binary dependence from the governor (the first token) to the dependent (the second token). The SBV relation, which stands for the *subject-predicate structure*, means that the head is a predicate verb and the dependent is a subject of the predicate verb; the VOB dependency relation, which stands for the *verb-object structure*, means that the head is a verb and the dependent is an object of the verb; the ADV relation, which stands for the *adverbial structure*, means that the head is a verb and the dependent is an adverb of the verb; the POB relation, which stands for the *prep-object structure*, means that the head is an object and the dependent is a preposition of the object.

Since  $V_{SBV} = V_{VOB} = V_t = \text{出生}$  (born) in this case, based on the predicate-argument model, the word “出生” should be extracted as a candidate event trigger.

## 2.2 Trigger Filter

Although we obtain some useful candidate triggers, certain meaningless candidate triggers come along in the results of the trigger extractor as well. Therefore, we introduce a trigger filter which uses heuristic rule and ranking algorithm to filter out these less informative candidates.

These rules are applied in order as follows:

### Rule (1): Subdividing Verbs

Since event trigger words are extracted based on the predicate-argument model, most of these candidate trigger words are verb terms. However, not all of verb terms can be used as trigger words. For example, the copular verb (e.g. “is”) rarely acts as the event trigger. To investigate which categories of verbs can serve as event triggers, we classify Chinese verbs into eight subclasses listed in Table 1. Such classification makes each subclass function as one grammatical role. For example, a modal verb will never be the predicate of a sentence and a nominal verb will always function as a noun.

We perform the verb sub-classification model based on the work by Liu et al. (2007). Statistically, about 94% of ACE Chinese event triggers are general verbs or nominal verbs and other types of verbs are rarely as trigger words. In order to ensure the accuracy of trigger clustering, we stress that the candidate trigger must be general verb or nominal verb.

### Rule (2): Domain Relevance Ranking

Verb	Description	Examples
vx	copular verb	他 <b>是</b> 对的 (He is right)
vz	modal verb	你 <b>应该</b> 努力工作 (You should work hard)
vf	formal verb	他 要求 <b>予以</b> 澄清 (He'd demand an explanation)
vq	directional verb	他 认识 <b>到</b> 困难 (He has realized the difficulties)
vb	resultative verb	他 看 <b>完</b> 了 电影 (He has seen the movie)
vg	<b>general verb</b>	他 <b>喜欢</b> 踢 足球 (He likes playing football)
vn	<b>nominal verb</b>	参加 我们的 <b>讨论</b> (Take part in our discussion)
vd	adverbial verb	产量 <b>持续</b> 增长 (Production increases steadily)

Table 1. The scheme of verb subclass

Domain relevancy degree is an important measure of the trigger’s significance. According to the candidate trigger distribution in the domain corpus and the general corpus, we can compute its domain relevancy degree as follows:

$$DR(V_i) = \text{Freq}_D(V_i) / \text{Freq}_G(V_i) \quad (1)$$

where  $DR(V_i)$  is the domain relevancy degree of the candidate trigger  $V_i$ ,  $\text{Freq}_D(V_i)$  is the frequency count of the candidate trigger  $V_i$  in the domain corpus (financial and musical news), and  $\text{Freq}_G(V_i)$  is the frequency count in the general corpus (People’s Daily corpus). We will rank candidate triggers by their domain relevancy degrees and retain top  $N_t^1$  candidate triggers.

## 2.3 Trigger Clustering and Event Type Paradigm Building

The trigger word is the most important lexical unit to represent events. A set of triggers with the same meaning and usage represents the same event type. Event type can be discovered based on trigger clustering. We propose the event type discovery (ETD) algorithm based on trigger clustering without giving the number of clusters in advance. The algorithm is shown in Figure 3.

For two triggers  $V_i$  and  $V_j$  in ETD, the similarity function  $\text{Sim}(V_i, V_j)$  in clustering is calculated using semantic information provided by HowNet (Dong et al., 2006) as

$$\text{Sim}(V_i, V_j) = \frac{2N_s}{N_i + N_j} \quad (2)$$

where  $N_s$  denotes the number of identical sememes in the DEFs (the concept definition in HowNet) of  $V_i$  and  $V_j$ ;  $N_i$  and  $N_j$  denote the number of sememes in the DEFs of  $V_i$  and  $V_j$ , respect-

<sup>1</sup> We test different  $N_t$  on dev set; and  $N_t$  is 50% of candidate triggers achieved the best gains.

<b>Algorithm 2: ETD algorithm</b>
<b>Input:</b> Candidate triggers from Section 2.2 and Threshold $\theta$ (refer to Section 3.2)
<b>Output:</b> Event trigger clusters
1: <b>ForEach</b> trigger $V_i$ in candidate triggers <b>Do</b>
2:   Compute the similarity ( $Sim$ ) between $V_i$ and the rest of other triggers, using function (2)
3: <b>If</b> $Sim \geq \theta$ <b>Then</b>
4:     add $V_i$ to the related event type $ET_{re} \cup \{V_i\}$
5: <b>Else If</b> $Sim < \theta$ <b>Then</b>
6:     set up a new event type $ET_{new}$
7: <b>End For</b>

Figure 3. The ETD algorithm

<b>Algorithm 3: PAC model</b>
<b>Input:</b> Verb-argument tuples $\langle V_i, Subj, Obj \rangle$ , where $V_i$ is the trigger from Section 2.2 and Subj and Obj are the arguments of $V_i$ ; and Threshold $\theta$ (refer to Section 3.2)
<b>Output:</b> Event trigger and arguments clusters
1: <b>ForEach</b> tuple $p$ in verb-argument tuples $\langle V_i, Subj, Obj \rangle$ <b>Do</b>
2:   Compute the similarity ( $Sim$ ) between $p$ and the rest of other tuples, using function (3) and (4)
3: <b>If</b> $Sim \geq \theta$ <b>Then</b>
4:     add $V_i$ to the related event type $ET_{re} \cup \{V_i\}$
5: <b>Else If</b> $Sim < \theta$ <b>Then</b>
6:     set up a new event type $ET_{new}$
7: <b>End For</b>

Figure 4. The PAC model

tively. Hownet uses sememes to interpret concepts. Sememes are regarded as the basic unit of the meaning. For example, “paper” can be viewed as a concept, and its sememes are “white”, “thin”, “soft”, “flammable”, etc.

As referred in Section 2.1, most of trigger words are verb terms. Polysemic verbs are a major issue in NLP, such as “to *fire* a gun” and “to *fire* a manager”, where “*fire*” has two different meanings. The state-of-the-art verb sense disambiguation approach (Wagner et al., 2009) stresses that verbs which agree on their selectional preferences belong to a common semantic class. For example, “to *arrest* the suspect” and “to *capture* the suspect”. Based on this approach, we propose a PAC (predicate-argument clustering) model which group the verbs based on their subcategorisation and selectional preferences. ETD considers only the verb subcategorisation, whereas PAC involves the verb argument tuple, such as  $\langle \text{bomb}, US\ Army, \text{weapon warehouse} \rangle$ , where “US Army” and “weapon warehouse” are the subject word and the object word of the trigger word “bomb”. The clustering process of PAC which is shown in Figure 4 is the same as ETD, except for the similarity measurement. PAC calculates the similarity between all the verb argument tuples by the following function:

$$Sim(Tuple_i, Tuple_j) = 2Sum_s / (Sum_i + Sum_j) \quad (3)$$

$$Sum_s = N_s + S_s + O_s, Sum_i = N_i + S_i + O_i, Sum_j = N_j + S_j + O_j \quad (4)$$

where  $S_s$  and  $O_s$  denotes the number of identical sememes in the DEFs of  $Subj_i$  and  $Subj_j$ ,  $Obj_i$  and  $Obj_j$ ;  $S_i$  and  $S_j$  denote the number of sememes in the DEFs of  $Subj_i$  and  $Subj_j$ , respectively;  $O_i$  and  $O_j$  denote the number of sememes in the DEFs of  $Obj_i$  and  $Obj_j$ , respectively.

A group of triggers are aggregated to a trigger cluster according to their semantic distance, and we view each trigger cluster as one kind of event type. Then all these event types are finally employed to construct an event type paradigm.

## 3 Experimental Results and Analysis

### 3.1 Experiment Settings

#### 3.1.1 Data Description

In order to test how robust our approach is, we evaluate it using three different data sets: ACE 05, Financial News<sup>2</sup> and Musical News<sup>3</sup>. ACE 05 is a public corpus with a pre-defined event type paradigm. Financial News and Musical News are specific domain corpora collected by ourselves. To justify the effectiveness of our method, we carefully conducted user studies into two specific domain corpora. For each sentence in the data, two annotators were asked to label and cluster all potential triggers. The agreement between our two annotators, measured using Cohen’s kappa coefficient, is substantial (kappa = 0.75). We asked the third annotator to adjudicate the trigger clusters on which the former two annotators disagreed. Each trigger cluster is used to represent one type of event. All these events construct our final event type paradigm. In particular, we carry out experiment on 633 documents from the ACE 05 corpus, 6000 sentences from the Financial News corpus and 6000 sentences from the Musical News corpus, respectively. One third of these data is used as development set and the remaining data is used as test set.

The gold standard event type paradigm of ACE, Financial News and Musical News are shown in Table 2, Table 3 and Table 4.

#### 3.1.2 Evaluation Measure

We adopt *F-Measure* ( $F$ ) and *Purity* (Halkidi et al., 2001) to determine the correctness of an event cluster:

$$p(i, r) = n(i, r) / n_r \quad (5)$$

<sup>2</sup> <http://www.10jqka.com.cn/>

<sup>3</sup> <http://yue.sina.com.cn/>



Method	Corpus	F-Measure (%)	Purity (%)
Baseline	ACE	42.05	61.47
ETD	ACE	63.21	68.17
PAC	ACE	<b>69.57</b>	<b>70.24</b>
ETD	Financial News	71.52	74.81
PAC	Financial News	<b>74.42</b>	<b>76.18</b>
ETD	Musical News	72.23	78.35
PAC	Musical News	<b>75.08</b>	<b>80.28</b>

Table 5. F-Measure and Purity scores on the test set. All the improvements are significant ( $p < 0.05$ )

Types	Subtype
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 2. ACE event type paradigm

Event Type	Examples
Start-Org	MIUI is found in 2010 by Xiaomi Tech.
End-Org	Sears closed more stores as holiday sales slide.
Merge-Org	Two of Tucson’s oldest and most respected landscape companies have decided to merge.
Declare Bankruptcy	American airlines are falling sharply for the second straight day on the fears that the company might be forced to file for bankruptcy.
Go-Public	Chinese video site Youku filed to go public on the New York Stock.
Raise-Price	Gold price rises higher in Hong Kong.
Cut-Price	Sony cuts Tablet S price by \$100, 16GB version now \$399.
Cooperation	Nokia and Microsoft announce plans for a broad strategic partnership to build a new global mobile ecosystem.
Investment	Tencent, one of the biggest web companies in China, is investing \$300m in Digital Sky Technologies of Russia.

Table 3. Financial News event type paradigm

Event Type	Examples
Vocal Concert	Chinese rock singer Cui Jian is to hold his first concert in Beijing at the Capital Gymnasium on Aug. 24.
Album	'Super Girls' release 1st album 'Terminal PK' on August 29, 2005.
Awards	Kanye West won best rap album at the 48th annual Grammy Awards in Los Angeles.
Sign-Org	Lady Gaga was signed with Streamline Records by the end of 2007.
Breakup-Org	Singer Chen Chusheng broke up with his agent E.E. Media after September.
Quit-Singing	Hong Kong pop queen and actress Faye Wong will soon quit her singing career.
Return-Stage	Faye Wong returned to the stage in 2010 amidst immense interest in the Sinosphere.

Table 4. Musical News event type paradigm

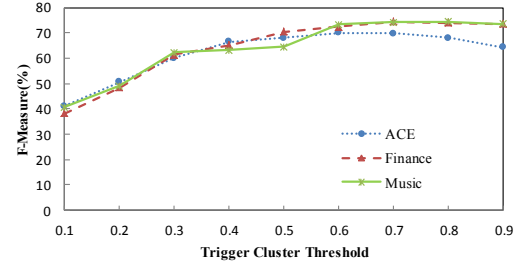


Figure 5. ETD algorithm with thresholding on the development set

$$r(i, r) = n(i, r) / n_i \quad (6)$$

$$f(i, r) = \frac{2 * p(i, r) * r(i, r)}{p(i, r) + r(i, r)} \quad (7)$$

$$F = \sum_i (n_i / n) \max \{f(i, r)\} \quad (8)$$

$$Purity = \sum_r (n_r / n) \max \{p(i, r)\} \quad (9)$$

where  $i$  is the gold standard event trigger cluster, and  $r$  is the event trigger cluster which has the most identical triggers with  $i$ . So  $n_i$  is the number of triggers in cluster  $i$ ;  $n_r$  is the number of triggers in cluster  $r$ ;  $n$  is the number of all triggers; and  $n(i, r)$  is the number of identical triggers between  $i$  and  $r$ . For every cluster we first compute  $p(i, r)$ ,  $r(i, r)$  and  $f(i, r)$ , then we obtain *F-Measure* and *Purity* for the whole clustering result. Note that the evaluation is based on word instances rather than word types.

### 3.2 Selection of Trigger Cluster Threshold

During development, we tuned the trigger clustering threshold to find the best value. Figure 5 presents the effect on F-Measure of varying the threshold for trigger clustering. This figure shows that the best performance on the development set can be obtained by selecting threshold 0.6 for the ACE corpus, 0.7 for the Financial News corpus and 0.9 for the Musical News corpus. Figure 5 suggests that the performance did not dramatically change with the threshold when  $\theta$  from 0.6 to 0.9. Hence, we can firstly set  $\theta = 0.6$  for new domains. We also test different threshold values for PAC, the result of which is the same as ETD. Then we directly use 0.6 as threshold value to the blind test.



### 3.3 Comparative Experiments

All the evaluation results are shown in Table 5. We *first* compare our approach with Li et al., 2010 (denoted as baseline) on the ACE 05 corpus. They exploit a cross-lingual clustering algorithm based on sentence-aligned bilingual parallel texts to discover event trigger clusters. The baseline approach can generate both English and Chinese event trigger clusters. We only compare with its Chinese result. Our approach achieves better performance than the baseline approach (about 8% significant improvement on Purity and more than 20% significant improvement on F-Measure). In addition, the baseline approach uses 1233 gold standard English event triggers and 852 gold standard Chinese event triggers in the ACE 05 as the input. However, we automatically extract event triggers based on our trigger extraction algorithm.

We carry out the *second* comparison experiment between ETD algorithm which is based on trigger clustering and PAC model which is based on predicate-argument clustering. The trigger and its corresponding arguments (selectional preferences) play an important role in our approach. We observe that the F-Measure score is boosted from 63.21% to 69.57% on the ACE corpus by using the PAC model. This can be explained by the reason that single trigger is not quite enough for representing event. Trigger’s arguments can contribute to trigger disambiguation. The experiment results also confirm the assumption (Wagner et al., 2009) that verbs which agree on their selectional preferences belong to a common semantic class.

We also run the *third* comparison experiment using three different corpora (ACE 05, Financial News and Musical News) to evaluate the robustness and domain adaptiveness of our system. The performances on the specific domain corpora are better than that on the ACE corpus (about 5% absolute improvement on F-Measure and 6%-10% on Purity). The main reason is that the events in specific domain are more specific. In addition, the experiment results on both specific domain corpora can achieve good performance. This indicates that our system is domain independent.

In order to evaluate whether the filter rules used in Section 2.2 are effective, we introduce the *fourth* comparison experiment. We use the Purity score to evaluate the effectiveness of the two filter rules. The evaluation results are shown in Table 6. We find that the average improvement using only rule (1) is 6.93% absolute for

Method	Performance		
	Purity (%)		
	ACE	Finance	Music
ETD	55.59	60.82	62.31
ETD+Rule(1)	65.13	66.51	69.37
ETD+Rule(2)	58.22	68.26	70.25
ETD+Rule(1)+Rule(2)	<b>68.17</b>	<b>74.81</b>	<b>78.35</b>
PAC	60.17	62.45	66.24
PAC+Rule(1)	68.04	69.24	72.38
PAC+Rule(2)	62.86	70.32	73.21
PAC+Rule(1)+Rule(2)	<b>70.24</b>	<b>76.18</b>	<b>80.28</b>

Table 6. The performance for filter rules

	Error types	Proportion
1	Trigger Extraction	33.0%
2	Trigger ambiguous	28.3%
3	Trigger Filter	19.5%
4	Others	19.2%

Table 7. Error types in the experiment

three corpora compared with the performance without using rule; using only rule (2) is 5.84% absolute; and using both rules, the average improvement is 12.61% absolute. This indicates that our two filter rules can improve the experiment performance significantly.

### 3.4 Discussion

#### Analysis of Experimental Errors

We first inspect the errors produced by our approach. The errors are mainly caused by the sparse event triggers in corpus. Table 7 shows the distribution of the errors in detail.

After error analysis, we found that the most number of errors are caused by trigger extraction. The main reasons are: firstly, not all of event triggers are verbs, such as “婚姻 (marriage)” for “Life/Marry” event, although it is reasonable to assume that event triggers are verbs because on average, there are more than 95% event triggers are verbs in our three different data sets. Secondly, since only verbs with subject and object are extracted, non-predicate verbs and the verbs without subject/object will not be extracted as candidate triggers. However, the coverage of possible triggers by our trigger extraction algorithm is reasonable good (more than 85%), because most of the trigger words appear repeatedly in the corpus, and their usages are varied. As long as one of their usages is fit for our extraction algorithm, they can be extracted as candidate triggers. Note that the goal of this paper is to build an event type paradigm for new domains. We concern more on the coverage of event type rather than event triggers. The event triggers extracted by us can cover all of event types. We will exploit more effective trigger extraction algorithm in future work.

Trigger ambiguity also accounts for a big proportion of the errors. As discussed in Section 2.3, we cannot judge the event type only by the trigger itself, such as “撤 (withdraw/dismiss)” for both “Personnel/End-Position” event and “Movement/Transport” event. This kind of errors can be partially fixed by the PAC model. For example, we cluster “撤职务 (dismiss duties)” for “Personnel/End-Position” event and “撤军队 (withdraw troop)” for “Movement/Transport” event. These examples indicate that selectional preferences seem to be a reasonable feature even for highly ambiguous verbs like “撤” which encourages to improve argument extraction.

There are still some errors caused by trigger filter. This is mainly due to the fact that not all of triggers are general verb or nominal verb. Domain relevance ranking filter rule will ignore the common event types, which might also be very important for general event extraction, such as “Life/Die” event in the ACE corpus. More effective filter rules will be exploited in future.

Some other errors are caused by NLP tools, such as word segmentation, part-of-speech tagging and dependency parsing. We believe that our algorithms can be improved with the improvement of these NLP tools. In addition, there are about 10% of good event triggers extracted but put into the wrong cluster by trigger cluster.

#### **Analysis of Different Corpus Sources**

The third comparison experiment shows that the performance of our approach on three corpora is not very consistent (F-Measure 69.57%, 74.42% and 75.08% on the ACE, Financial and Musical corpus, respectively). The F-Measure on the ACE corpus is lower than that on the other two domain corpora. The performances on the other two domain corpora are comparable. The main reasons are as follows: firstly, the discrimination between some event types in the ACE paradigm is very small, such as the “Justice/Charge-Indict” event and the “Justice/Sue” event; the “Personnel/Nominate” event and the “Personnel/Start-Position” event; the “Life/Die” event and the “Conflict/Attack” event. Secondly, some events rarely occur in the ACE corpus, such as “Justice/Extradite” event occurs only three times in the ACE corpus. Thirdly, some events have a lot of triggers in the ACE corpus, but not all of these event triggers appear frequently. For example, the “Movement/Transport” event has 188 triggers and 64.89% of its triggers appear only once. As compared to ACE corpus, the similarity among event types in the other two

corpora is low. Finally, we analyze that the quantity of event types also results in the different performance between the ACE corpus and the domain-specific corpus. There are 33 subtypes of events in the ACE corpus which are far more than the number of events in the Financial and Musical corpus.

#### **Analysis of Different Filter Rules**

The fourth comparison experiment indicates that both the filter rules are effective. As shown in Table 4, the improvement obtained using rule (1) is 7.87%, 6.79% and 6.14% on the ACE, Financial and Musical News corpus, respectively. The experiment result verifies that verb subdividing is helpful for the Chinese event extraction task. The improvement obtained using rule (2) is 2.69%, 7.87% and 6.97% on the ACE, Financial and Musical News corpus, respectively. The performances on all these three different corpora are improved by rule (2); however, it is obvious that rule (2) is not much effective on the ACE corpus (2.69%) compared with the other two domain-specific corpora (7.87% and 6.97%). The main reason is that the ACE corpus contains many common events and the domain-specific information is not very useful. For the other two domain-specific corpora, rule (2) has improved the performance more than rule (1) did. This is due to the fact that rule (2) is more effective on the domain-specific corpus.

## **4 Related Work**

### **4.1 Word Cluster Discovery**

Our approach of automatically building an event type paradigm is related to some prior work on word cluster discovery (e.g. Barzilay and McKeown, 2001; Ibrahim et al., 2003; Pang et al., 2003). Most of these works are based on machine translation techniques to solve paraphrase extraction problem. However, several recent researches have stressed the benefits of using word clusters to improve the performance of information extraction tasks. For example, Miller et al., (2004) proved that word clusters could significantly improve English name tagging performance. In the same vein, some studies work on the problem of relation extraction (Chambers and Jurasky, 2011 and 2009; Poon and Domingos, 2009 and 2008; Yates and Etzioni, 2009). In these work, “relation words” were extracted and clustered. In this paper, our work confirmed that trigger clusters are also effective for event type paradigm building. The problem of event trigger

words extraction and clustering is also a challenge problem.

#### 4.2 Traditional Event Extraction

The commonly used approaches for most event extraction systems are the knowledge engineering approach and the machine learning approach. Grishman et al., (2005) used a combination of pattern matching and statistical modeling techniques. They extract two kinds of patterns: 1) the sequence of constituent heads separating anchor and its arguments; and 2) a predicate argument sub-graph of the sentence connecting anchor to all the event arguments. In conjunction, they used a set of Maximum Entropy based classifiers for 1) Trigger labeling, 2) Argument classification and 3) Event classification. Ji and Grishman, (2008) further exploited a correlation between senses of verbs (that are the triggers for events) and topics of documents. They first proposed refining event extraction through unsupervised cross-document inference. Following Ji's work, Liao et al., (2010) used document level cross-event inference to improve event extraction. Chen and Ji, (2009) combined word-based classifier with character-based classifier; and explored effective features for the Chinese event extraction task. Liao and Grishman, (2010) ranked two semi-supervised learning methods for adapting the event extraction system to new event types. Hong et al, (2011) proposed a blind cross-entity inference method for event extraction, which well uses the consistency of entity mention to achieve sentence-level trigger and argument (role) classification. Lu and Roth, (2012) presented a novel model based on the semi-Markov conditional random fields for the challenging event extraction task. The model takes in coarse mention boundary and type information and predicts complete structures indicating the corresponding argument role for each mention.

However, for all the above approaches, it is necessary to specify the target event type in advance. Defining and identifying those types heavily rely on expert knowledge, and reaching an agreement among the experts or annotators requires a lot of human labor. Li et al., (2010) proposed a domain-independent novel event discovery approach. They exploited a cross-lingual clustering algorithm based on sentence-aligned bilingual parallel texts to discover event trigger clusters. Their motivation is to discover novel events for a new domain rather than build a new event type paradigm from scratch. Therefore, it takes domain specific event triggers as the input.

However, it is also a costly task to annotate triggers for new domains. The motivation of this paper is to build event type paradigm from scratch rather than discover novel events based on the existing event type paradigm.

#### 5 Conclusion and Future Work

Traditionally, in the topic of event detection, we have to categorize the events into various pre-defined event-types. In this paper, we aim to tackle the situation when the category of event-type is undefined, and we try to derive the event-types from the corpus. In particular, we automatically build an event type paradigm by using a trigger clustering algorithm: 1) we introduce a trigger extraction algorithm based on the dependency syntactic structure; 2) a trigger filter is then constructed to remove some noisy candidate triggers; 3) we develop an event type discovery algorithm based on our proposed trigger clustering methods. The clustered event types are used to construct an event type paradigm. Experimental results on three different corpora – ACE (small, homogeneous, open corpus), Financial News and Musical News (large scale, specific domain, web corpus) indicate that our method can effectively build an event type paradigm and that it is easy to adapt the proposed method to new domains.

In the future, more sophisticated algorithm will be exploited. Furthermore, a bottom-up event extraction system can be built based on our event type paradigm.

#### Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 61273321, 61133012 and the National 863 Leading Technology Research Project via grant 2012AA011102.

#### References

- R. Barzilay and K. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of ACL/EACL*.
- N. Chambers and D. Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of ACL-IJCNLP*.
- N. Chambers and D. Jurafsky. 2011. Template-Based Information Extraction without the Templates. In *Proceedings of ACL*.

- W. Che, Z. Li, Y. Li, Y. Guo, B. Qin, T. Liu. 2009. Multilingual dependency-based syntactic and semantic parsing. In *Proceedings of CoNLL 2009*.
- H. Chieu and H. Ng. 2002. A Maximum Entropy Approach to Information Extraction from Semi-structured and Free Text. In *Proceedings of AAAI*.
- N. Chinchor, L. Hirschman and D. D. Lewis. 1994. Evaluating Message Understanding Systems: An Analysis of the Third Message Understanding Conference (MUC-3). In *Computational Linguistics 3*, pages 409-449.
- X. Ding, F. Song, B. Qin and T. Liu. 2011. Research on Typical Event Extraction Method in the Field of Music. *Journal of Chinese Information Processing Vol. 25, No. 2, 15-20*.
- Z. Dong and Q. Dong. 2006. HowNet and the Computation of Meaning. World Scientific Publishing Co. Pte. Ltd. 2006.
- R. Grishman, D. Westbrook and A. Meyers. 2005. Nyu's English ACE 2005 System Description. In *Proceedings of NIST 2005*.
- M. Halkidi, Y. Batistakis, M. Vazirgiannis. On Clustering Validation Techniques[J]. *Intelligent Information Systems 2001*. 17( 2-3): 107-145.
- T. Hasegawa, S. Sekine and R. Grishman. 2006. Discovering Relations among Named Entities from Large Corpora. In *Proceedings of ACL*.
- Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, Q. Zhu. 2011. Using Cross-Entity Inference to Improve Event Extraction. In *Proceedings of ACL*.
- A. Ibrahim, B. Katz and J. Lin. 2003. Extracting Structural Paraphrases from Aligned Monolingual Corpora. In *Proceedings of the Second International Workshop on Paraphrasing (IWP 2003)*.
- H. Ji and R. Grishman. 2008. Refining Event Extraction through Unsupervised Cross-document Inference. In *Proceedings of ACL*.
- LDC. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events. [http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines\\_v5.4.3.pdf](http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines_v5.4.3.pdf)
- C. S. Lee, Y. J. Chen, and Z. W. Jian, Ontology-based fuzzy event extraction agent for chinese e-news summarization. *Expert Systems With Applications*, vol. 25, no. 3, pp. 431-447, 2003.
- H. Li, X. Li, H. Ji, and Y. Marton. Domain-Independent Novel Event Discovery and Semi-Automatic Event Annotation. In *Proceedings of PACLIC*.
- S. Liao and R. Grishman. 2010. Filtered Ranking for Bootstrapping in Event Extraction. In *Proceedings of Coling*.
- T. Liu, J. Ma, H. Zhang and S. Li. 2007. Subdividing Verbs to Improve Syntactic Pasing. *Journal of Electronics*, 24(3): 347-352.
- W. Lu and D. Roth. 2012. Automatic Event Extraction with Structured Preference Modeling. In *Proceedings of ACL*.
- S. Miller, J. Guinness and A. Zamanian. 2004. Name Tagging with Word Clusters and Discriminative Training. In *Proceedings of HLT/ACL*.
- B. Pang, K. Knight and D. Marcu. 2003. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. In *Proceedings of NAACL*.
- H. Poon and P. Domingos. 2008. Joint Unsupervised Coreference Resolution with Markov Logic. In *Proceedings of EMNLP*.
- H. Poon and P. Domingos. 2009. Unsupervised Semantic Parsing. In *Proceedings of EMNLP*.
- B. Rosenfeld and R. Feldman. 2006. URES: an Unsupervised Web Relation Extraction System. In *Proceedings of the COLING/ACL*.
- F. Schilder. 2007. Event Extraction and Temporal Reasoning in Legal Documents. In *Proceedings of Annotating, Extracting and Reasoning about Time*, LNAI-Volume 4795/2007, pp: 59-71.
- K. Sudo, S. Sekine, and R. Grishman. 2001. Automatic pattern acquisition for Japanese information extraction. In *Proceedings of HLT*.
- K. Sudo, S. Sekine, and R. Grishman. 2003. An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of ACL*.
- M. Thelen and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. In *Proc. of EMNLP*.
- W. Wagner, H. Schmid, and S. Schulte im Walde. 2009. Verb Sense Disambiguation using a Predicate Argument-Clustering Model. In *Proceedings of the CogSci Workshop on Distributional Semantics beyond Concrete Concepts*.
- A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii, 2001. Event extraction from biomedical papers using a full parser. *Pacif. Symp. Biocomp*, 6, pp: 408-419.
- M. Yankova 2003. Focusing on Scenario Recognition in Information Extraction. In *Proceedings of ECAL*.
- A. Yates and O. Etzioni. 2009. Unsupervised Methods for Determining Object and Relation Synonyms on the Web. *Journal of Artificial Intelligence Research 34 (2009) 255-296*.
- Q. Zhou and M. Sun. 1999. Build a Chinese treebank as the test suite for Chinese parsers. In *Proceedings of the workshop MAL '99 and NLPRS '99*.

# Chinese Named Entity Abbreviation Generation Using First-Order Logic

Huan Chen, Qi Zhang, Jin Qian, Xuanjing Huang

School of Computer Science

Fudan University

Shanghai, P.R. China

{12210240054, qz, 12110240030, xjhuang}@fudan.edu.cn

## Abstract

Normalizing named entity abbreviations to their standard forms is an important preprocessing task for question answering, entity retrieval, event detection, microblog processing, and many other applications. Along with the quick expansion of microblogs, this task has received more and more attentions in recent years. In this paper, we propose a novel entity abbreviation generation method using first-order logic to model long distance constraints. In order to reduce the human effort of manual annotating corpus, we also introduce an automatically training data construction method with simple strategies. Experimental results demonstrate that the proposed method achieves better performance than state-of-the-art approaches.

## 1 Introduction

Twitter and other social media services have received considerable attentions in recent years. Users provide hundreds of millions microblogs through them everyday. The informative data has been relied on by many applications, such as sentiment analysis (Jiang et al., 2011; Meng et al., 2012), event detection (Sakaki et al., 2010; Lin et al., 2010), stock market predication (Bollen et al., 2011), and so on. However, due to the constraint on the length of characters, abbreviations frequently occur in microblogs. According to a statistic, approximately 20% of sentences in news articles have abbreviated words (Chang and Lai, 2004). The frequency of abbreviation has become even more popular along with the rapid increment of user generated contents. Without pre-normalizing these abbreviations, most of the natural language processing systems may heavily suffer from them.

The goal of entity abbreviation generation is to produce abbreviated equivalents of the original entities. Table 1 shows several examples of entities and their corresponding abbreviations. A few of approaches have been done on this task. Li and Yarowsky (Li and Yarowsky, 2008b) introduced an unsupervised method used to extract phrases and their abbreviation pair using parallel dataset and monolingual corpora. Xie et al. (2011) proposed to use weighted bipartite graph to extract definition and corresponding abbreviation pairs from anchor texts. Since these methods rely heavily on lexical/phonetic similarity, substitution of characters and portion may not be correctly identified through them. Yang et al. (2009) studied the Chinese entity name abbreviation problem. They formulated the abbreviation task as a sequence labeling problem and used the conditional random fields (CRFs) to model it. However the long distance and global constraint can not be easily modeled thorough CRFs.

Entity	Abbr.
北京大学 (Peking University)	北大
中国石油天然气集团公司 (China National Petroleum Corporation)	中石油
中国国际航空公司 (Air China)	国航

Table 1: Abbreviation examples

To overcome these limitations, in this paper, we propose a novel entity abbreviation generation method, which combines first-order logic and rich linguistic features. To the best of our knowledge, our approach is the first work of using first-order logic for this entity abbreviation. Abbreviation generation is converted to character deletion and

keep operations which are modeled by logic formula. Linguistic features and relations between different operations are represented by local and global logic formulas respectively. Markov Logic Networks (MLN) (Richardson and Domingos, 2006) is adopted for learning and predication. To reduce the human effort in constructing the training data, we collect standard forms of entities from online encyclopedia and introduce a few of simple patterns to extract abbreviations from documents and search engine snippets with high precision as training data. Experimental results show that the proposed methods achieve better performance than state-of-the-art methods and can efficiently process large volumes of data.

The remainder of the paper is organized as follows: In section 2, we review a number of related works and the state-of-the-art approaches in related areas. Section 3 presents the proposed method. Experimental results in test collections and analyses are shown in section 4. Section 5 concludes this paper.

## 2 Related Work

The proposed approach builds on contributions from two research communities: text normalization, and Markov Logic Networks. In the following of this section, we give brief description of previous works on these areas.

### 2.1 Text Normalization

Named entity normalization, abbreviation generation, and lexical normalization are related to this task. These problems have been recognized as important problems for various languages. Since different languages have their own peculiarities, many approaches have been proposed to handle variants of words (Aw et al., 2006; Liu et al., 2012; Han et al., 2012) and named entities (Yang et al., 2009; Xie et al., 2011; Li and Yarowsky, 2008b).

Chang and Teng (2006) introduced an HMM-based single character recovery model to extract character level abbreviation pairs for textual corpus. Okazaki et al. (2008) also used discriminative approach for this task. They formalized the abbreviation recognition task as a binary classification problem and used Support Vector Machines to model it. Yang et al. (2012) also treated the abbreviation generation problem as a labeling task and used Conditional Random Fields (CRFs) to do it. They also proposed to re-rank candidates by a

length model and web information.

Li and Yarowsky (2008b) proposed an unsupervised method extracting the relation between a full-form phrase and its abbreviation from monolingual corpora. They used data co-occurrence intuition to identify relations between abbreviation and full names. They also improved a statistical machine translation by incorporating the extracted relations into the baseline translation system. Based on the data co-occurrence phenomena, they introduced a bootstrapping procedure to identify formal-informal relations informal phrases in web corpora (Li and Yarowsky, 2008a). They used search engine to extract contextual instances of the given an informal phrase, and ranked the candidate relation pairs using conditional log-linear model. Xie et al. (2011) proposed to extract Chinese abbreviations and their corresponding definitions based on anchor texts. They constructed a weighted URL-AnchorText bipartite graph from anchor texts and applied co-frequency based measures to quantify the relatedness between two anchor texts.

For lexical normalisation, Aw et al. (2006) treated the lexical normalisation problem as a translation problem from the informal language to the formal English language and adapted a phrase-based method to do it. Han and Baldwin (2011) proposed a supervised method to detect ill-formed words and used morphophonemic similarity to generate correction candidates. Liu et al. (2012) proposed to use a broad coverage lexical normalization method consisting three key components enhanced letter transformation, visual priming, and string/phonetic similarity. Han et al. (2012) introduced a dictionary based method and an automatic normalisation-dictionary construction method. They assumed that lexical variants and their standard forms occur in similar contexts.

In this paper, we focused on named entity abbreviation generation problem and treated the problem as a labeling task. Due to the flexibilities of Markov Logic Networks on capturing local and global linguistic feature, we adopted it to model the supervised classification procedure. To reduce the human effort in constructing training data, we also introduced a sample rule based method to find relations between standard forms and abbreviations.

<b>Predicates about characters in the entity</b>	
<i>character(i,c)</i>	The <i>i</i> th character is <i>c</i> .
<i>isNumber(i)</i>	The <i>i</i> th character is a number.
<b>Predicates about words in the entity</b>	
<i>word(j,w)</i>	The <i>j</i> th word is <i>w</i> .
<i>isCity(j)</i>	The <i>j</i> th word is a city name.
<i>lastWord(j)</i>	The <i>j</i> th word is the last word.
<i>sufCorp(j)</i>	The <i>j</i> th word belongs the set of common suffixes of corporation.
<i>sufSchool(j)</i>	The <i>j</i> th word belongs the set of common suffixes of school.
<i>sufOrg(j)</i>	The <i>j</i> th word belongs the set of common suffixes of organizations.
<i>sufGov(j)</i>	The <i>j</i> th word belongs the set of common suffixes of government agencies.
<i>idf(j,v)</i>	The inverse document frequency of <i>j</i> th word is <i>v</i> .
<b>Predicates about entire entity</b>	
<i>entityType(t)</i>	The type of the entity is <i>t</i> .
<i>lenChar(n)</i>	The total number of characters is <i>n</i> .
<i>lenWord(n)</i>	The total number of words is <i>n</i> .
<b>Predicates about relations between characters and words</b>	
<i>cwMap(i,j)</i>	The <i>i</i> th character belongs to <i>j</i> th word.
<i>cwPosition(i,j)</i>	The <i>i</i> th character of the entity is the <i>j</i> th character in the corresponding word.

Table 2: Descriptions of observed predicates.

## 2.2 Markov Logic Networks

Richardson and Domingos (2006) proposed Markov Logic Networks (MLN), which combines first-order logic and probabilistic graphical models. MLN framework has been adopted for several natural language processing tasks and achieved a certain level of success (Singla and Domingos, 2006; Riedel and Meza-Ruiz, 2008; Yoshikawa et al., 2009; Andrzejewski et al., 2011; Jiang et al., 2012; Huang et al., 2012).

Singla and Domingos (2006) modeled the entity resolution problem with MLN. They demonstrated the capability of MLN to seamlessly combine a number of previous approaches. Poon and Domingos (2008) proposed to use MLN for joint unsupervised coreference resolution. Yoshikawa et al. (2009) proposed to use Markov logic to incorporate both local features and global constraints that hold between temporal relations. Andrzejewski et al. (2011) introduced a framework for incorporating general domain knowledge, which is represented by First-Order Logic (FOL) rules, into LDA inference to produce topics shaped by both the data and the rules.

## 3 The Proposed Approach

In this section, firstly, we briefly describe the Markov Logic Networks framework. Then, we present the first-order logic formulas including local formulas and global formulas we used in this work.

### 3.1 Markov Logic Networks

A MLN consists of a set of logic formulas that describe first-order knowledge base. Each formula consists of a set of first-order predicates, logical connectors and variables. Different with first-order logic, these hard logic formulas are softened and can be violated with some penalty (the weight of formula) in MLN.

We use  $\mathcal{M}$  to represent a MLN and  $\{(\phi_i, w_i)\}$  to represent formula  $\phi_i$  and its weight  $w_i$ . These weighted formulas define a probability distribution over sets of possible worlds. Let  $y$  denote a possible world, the  $p(y)$  is defined as follows (Richardson and Domingos, 2006):

$$p(y) = \frac{1}{Z} \exp \left( \sum_{(\phi_i, w_i) \in \mathcal{M}} w_i \sum_{c \in C^{n_{\phi_i}}} f_c^{\phi_i}(y) \right),$$

where each  $c$  is a binding of free variable in  $\phi_i$  to constraints;  $f_c^{\phi_i}(y)$  is a binary feature function that

returns 1 if the true value is obtained in the ground formula we get by replacing the free variables in  $\phi_i$  with the constants in  $c$  under the given possible world  $y$ , and 0 otherwise;  $C^{n_{\phi_i}}$  is all possible bindings of variables to constants, and  $Z$  is a normalization constant.

Many methods have been proposed to learn the weights of MLNs using both generative and discriminative approaches (Richardson and Domingos, 2006; Singla and Domingos, 2006). There are also several MLNs learning packages available online such as thebeast<sup>1</sup>, Tuffy<sup>2</sup>, PyMLNs<sup>3</sup>, Alchemy<sup>4</sup>, and so on.

### 3.2 MLN for Abbreviation Generation

In this work, we convert the abbreviation generation problem as a labeling task for every characters in entities. Predicate  $drop(i)$  indicates that the character at position  $i$  is omitted in the abbreviation. Previous works (Chang and Lai, 2004; Yang et al., 2009) show that Chinese named entities can be further segmented into words. Words also provide important information for abbreviation generation. Hence, in this work, we also segment named entities into words and propose an observed predict to connect words and characters.

#### 3.2.1 Local Formulas

The local formulas relate one or more observed predicates to exactly one hidden predicate. In this work, we define a list of observed predicates to describe the properties of individual characters. Table 2 shows the list. For this task, there is only one hidden predicate  $drop$ .

Table 3 lists the local formulas used in this work. The “+” notation in the formulas indicates that the each constant of the logic variable should be weighted separately. For example, formula  $character(2, \bar{\quad}) \wedge isNumber(2) \Rightarrow drop(2)$  and  $character(2, +) \wedge isNumber(2) \Rightarrow drop(2)$  may have different weights as inferred by formula  $character(i, c+) \wedge isNumber(i) \Rightarrow drop(i)$ .

Three kinds of local formulas are introduced in this work. Lexical features are used to capture the context information based on both character and word level information. Distance and position features are helpful in determining which parts of a entity may be removed. Hence, we

also incorporate position information of word into local formulas. For example, “大学(University)” is usually omitted when it is at the end of the entity. In practice, abbreviations of some kinds of entities can be generated through several strategies. So we introduce several local formulas to handle a group of related entities with similar suffix.

#### 3.2.2 Global Formulas

Global formulas are designed to handle deletion of multiple characters. Since in this work, we only have one hidden predicate,  $drop$ , the global formulas incorporate correlations among different ground atoms of the  $drop$  predicate.

We propose to use global formulas to force the abbreviations to contain at least 2 characters and to make sure that at least one character is deleted. The following formulas are implemented:

$$\begin{aligned} &|character(i, c) \wedge drop(i)| \text{ all } i \geq 1 \\ &|character(i, c) \wedge \neg drop(i)| \text{ all } i \geq 2 \end{aligned}$$

Another constraint is that for the characters in some particular words should by dropped or kept simultaneously. So we add two formulas to model this:

$$\begin{aligned} &character(i, c1) \wedge cwMap(i, j) \wedge drop(i) \wedge \\ &character(i + 1, c2) \wedge cwMap(i + 1, j) \\ &\Rightarrow drop(i + 1) \end{aligned}$$

$$\begin{aligned} &character(i, c1) \wedge cwMap(i, j) \wedge \neg drop(i) \wedge \\ &character(i + 1, c2) \wedge cwMap(i + 1, j) \\ &\Rightarrow \neg drop(i + 1) \end{aligned}$$

## 4 Experiments

In this section, we first describe the dataset construction method, evaluation metrics, and experimental configurations. We then describe the evaluation results and analysis.

### 4.1 Data Set

For training and evaluating the performance the proposed method, we need a large number of abbreviation and corresponding standard form pairs. However, manually labeling is a laborious and time consuming work. To reduce human effort, we propose to construct annotated dataset with two steps. Firstly, we collect entities from Baidu

<sup>1</sup><http://code.google.com/p/thebeast>

<sup>2</sup><http://hazy.cs.wisc.edu/hazy/tuffy/>

<sup>3</sup><http://www9-old.in.tum.de/people/jain/mlns/>

<sup>4</sup><http://alchemy.cs.washington.edu/>



---

**Lexical Features**

$\text{character}(i,c+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{isNumber}(i) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{word}(j,w+) \wedge \text{cwMap}(i,j) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{word}(j,w+) \wedge \text{cwMap}(i,j) \wedge \text{lastWord}(j) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{word}(j,w+) \wedge \text{cwMap}(i,j) \wedge \text{idf}(j,v+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{word}(j,w+) \wedge \text{cwMap}(i,j) \wedge \text{entity}(e+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{word}(j,w+) \wedge \text{cwMap}(i,j) \wedge \text{isCity}(j) \wedge \text{entityType}(e+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{word}(j,w+) \wedge \text{cwMap}(i,j) \wedge \text{isCity}(j) \wedge \text{word}(j,w1+) \wedge \text{word}(j+1,w2+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j-1,w+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+2,w+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+3,w+) \Rightarrow \text{drop}(i)$

---

**Distance and Position Features**

$\text{character}(i,c) \wedge \text{lenWord}(wn+) \wedge \text{cwPosition}(i,wp+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{lenChar}(cn+) \wedge \text{cwPosition}(i,wp+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j,w+) \wedge \text{cwPosition}(i,wp+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j,w+) \wedge \text{lenWord}(wn+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \wedge \text{cwPosition}(i,wp+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \wedge \text{cwPosition}(i,wp+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+2,w+) \wedge \text{cwPosition}(i,wp+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+2,w+) \wedge \text{cwPosition}(i,wp+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+3,w+) \wedge \text{cwPosition}(i,wp+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+3,w+) \wedge \text{cwPosition}(i,wp+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$

---

**Features for Entity with Special Suffixes**

$\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j,w+) \wedge \text{lenWord}(l+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{isCity}(j) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{isCity}(j) \wedge \text{cwMap}(i,j) \wedge \text{word}(j,w1+) \wedge \text{word}(j+1,w2+) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwPosition}(i,p+) \wedge \neg \text{isCity}(j) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \wedge$   
 $\quad (\text{sufSchool}(j+1) \vee \text{sufOrg}(j+1) \vee \text{sufGov}(j+1)) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \wedge (\text{sufSchool}(j+1) \vee \text{sufOrg}(j+1) \vee \text{sufGov}(j+1)) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j-1,w+) \wedge (\text{sufSchool}(j) \vee \text{sufOrg}(j) \vee \text{sufGov}(j)) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j-2,w+) \wedge (\text{sufSchool}(j) \vee \text{sufOrg}(j) \vee \text{sufGov}(j)) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c+) \wedge \text{cwMap}(i,j) \wedge \text{word}(j,w1+) \wedge \text{cwMap}(ip,j-1) \wedge \text{city}(ip,p) \wedge$   
 $\quad (\text{sufSchool}(j+1) \vee \text{sufOrg}(j+1) \vee \text{sufGov}(j+1)) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j,w+) \wedge \text{entityType}(t+) \wedge \neg \text{isCity}(j) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+1,w+) \wedge \text{entityType}(t+) \wedge \neg \text{isCity}(j) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+2,w+) \wedge \text{entityType}(t+) \wedge \neg \text{isCity}(j) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+3,w+) \wedge \text{entityType}(t+) \wedge \neg \text{isCity}(j) \Rightarrow \text{drop}(i)$   
 $\text{character}(i,c) \wedge \text{cwMap}(i,j) \wedge \text{word}(j+4,w+) \wedge \text{entityType}(t+) \wedge \neg \text{isCity}(j) \Rightarrow \text{drop}(i)$   
 $\text{cwMap}(i,j) \wedge \text{word}(j-1,w+) \wedge \text{isCity}(j-1) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$   
 $\text{cwMap}(i,j) \wedge (j=0) \wedge \text{word}(j,w) \wedge \text{entityType}(t+) \Rightarrow \text{drop}(i)$

---

Table 3: Descriptions of local formulas.

简称: A (abbreviation name: A)
A是E(的)?(中文)?简称 (A is the (Chinese)? abbreviation name of E)
E(的)?(中文)?简称(是)?A (the (Chinese)? abbreviation name of E is A)
A是E的同义词 (A is a synonym of E)
E和A是同义词 (E and A are synonyms)
A和E是同义词 (A and E are synonyms)
E represents the entity
A represents the candidate abbreviation

Table 4: The lexical level regular expressions used to match entity and abbreviation pairs.

Baike<sup>5</sup>, which is one of the most popular wiki-based Chinese encyclopedia and contains more than 6 millions items. Secondly, we use several simple regular expressions to extract abbreviation of entities from the crawled encyclopedia and snippets of search engine.

We crawled 3.2 millions articles from Baidu Baike. After that, we cleaned the HTML tags and extracted title, category and textual content from each articles. the structure of Baidu Baike is similar to that of Wikipedia, where titles are the name of the subject of the article, or may be a description of the topic. Hence, titles can be considered as the standard forms of entities. We select titles whose categories belong to location, organization, and facility to construct the standard forms list. It contains 302,633 items in total.

The next step is to use titles and corresponding articles to extract abbreviations. Through analyzing the dataset, we observe that most of abbreviations with the explicit description can be matched through a few of lexical level regular expressions. Table 4 shows the regular expressions we used in this work. Through this step, 30,701 abbreviation and entity pairs are extracted. We randomly select 500 pairs from them and manually check their correctness. The accuracy of the extracted pairs is around 98.2%.

To further increase the number of extractions, we propose to use Web as corpus and extract abbreviation and entity pairs from snippets of

search engine results. For each entity whose abbreviation cannot be identified through the regular expressions described above, we combine entity and “简称 (abbreviation)” as queries for retrieving. The first three regular expressions in Table 4 are used to match abbreviation and entity pairs. Through this step, we get another 19,531 abbreviations. We also randomly select 500 pairs from them and manually check their correctness. The accuracy is around 95.2%. Finally, we merge the pairs extracted from Baike and search engine snippets and construct a list containing 50,232 abbreviation entity pairs. The accuracy of the list is 97.03%.

## 4.2 Experimental Settings

For evaluating the performance of the proposed method, we conducted experiments on the automatical constructed data. Total instances are randomly split with 75% for training, 5% for development and the other 20% for testing.

We compare the proposed method against start-of-the-art systems. Yang et al. (2009) proposed to use CRFs to model this. In this work, firstly, we re-implement the features they proposed. To fairly compare the two models, we also extend their work by including all local formulas we used in this work as features.

In our setting, we use FudanNLP<sup>6</sup> toolkit and thebeast<sup>7</sup> Markov Logic engine. FudanNLP is developed for Chinese natural language processing. We use the Chinese word segmentation of it under the default settings. The detailed setting of thebeast engine is as follows: The inference algorithm is the MAP inference with a cutting plane approach. For parameter learning, the weights for formulas is updated by an online learning algorithm with MIRA update rule. All the initial weights are set to zeros. The number of iterations is set to 10 epochs.

For evaluation metrics, we use precision, recall, and F-score to evaluate the performance of character deletion operation. To evaluate the performance of the entire generated abbreviations, we also propose to use accuracy to do it. It means that the generated abbreviation is considered as correct if all characters of its standard form are correctly classified.

<sup>5</sup><http://baike.baidu.com>

<sup>6</sup><http://code.google.com/p/fudannlp>

<sup>7</sup><http://code.google.com/p/thebeast>

Methods	P	R	F	A
MLN-LF	83.2%	81.1%	82.1%	42.2%
MLN-LF+DPF	80.9%	84.3%	82.6%	45.7%
MLN-Local	82.4%	85.4%	83.9%	54.7%
MLN-Local+Global	81.6%	85.9%	83.7%	56.8%
CRFs-Yang	82.9%	83.6%	83.2%	39.7%
CRFs-LF	84.9%	83.7%	84.3%	40.5%
CRFs-LF+DPF	85.5%	83.5%	84.5%	40.6%
CRFs-Local	84.9%	83.8%	84.3%	40.8%

Table 5: The lexical level regular expressions used to match entity and abbreviation pairs.

### 4.3 Results

To evaluate the performance of our method, we set up several variants of the proposed method to compare with performances of CRFs. The *MLN-LF* method uses only the lexical features described in the Table 3. The *MLN-LF+DPF* method uses both lexical features and distance and position features. The *MLN-Local* method uses all local formulas described in the Table 3. The *MLN-Local+Global* methods combine both local formulas and global formulas together. For Yang’s system, we use *CRFs-Yang* to represent the re-implemented method with feature set proposed by them and *CRFs-LF*, *CRFs-LF+DPF*, and *CRFs-Local* to represent feature sets similar as used by MLN.

Table 5 shows the performances of different methods. We can see that *MLN-Local+Global* achieve the best accuracy of entire abbreviation among all the methods. Although, the F-score of *MLN-Local+Global* is slightly worse than *MLN-Local*. We think that the global formulas contribute a lot for the entire accuracy. However, since the constraint of simultaneously dropping or keeping characters does not consider context, it may also bring some false matches. We can also see that, the methods modeled by MLN significantly outperform the performances of CRFs no matter which feature sets are used (based on a paired 2-tailed t-test with  $p < 0.05$ ). We think that overfitting may be one of the main reasons.

From the perspective of entire accuracy, comparing the performances of *MLN-LF+DPF* and *MLN-Local*, we can see that features for entities with special suffixes contribute a lot. The relative improvement of *MLN-Local* is around 19.7%. It shows that the explicit rules are useful for improv-

ing the performance. However, these explicit rules only bring a small improvement to the accuracy of CRFs.

Comparing the performances of CRFs and MLNs, we can observe that CRFs achieve slightly better performance in classifying single characters. However MLNs achieve significantly better results of the entire accuracies. We think that these kinds of long distance features can be well handled by MLNs. These features are useful to capture the global constraints. Hence, MLNs can achieve better accuracy of the entire abbreviations.

In this paper, we also investigate the performance of different methods as the training data size are varied. Figure 1 shows the results. All full lines show the results of MLNs with different feature sets. The dot dash lines show the results of CRFs. From the results, we can observe that MLNs perform better than CRFs in most of cases. Except that MLNs with only lexical features work slightly worse than CRFs with small number of training data. From the figure, we also observe that the performance improvement of CRFs are not significant when the number of training data is larger than 35,000. However, methods using MLNs benefit a lot from the increasing data size. If more training instances are given, the performance of MLNs can still be improved.

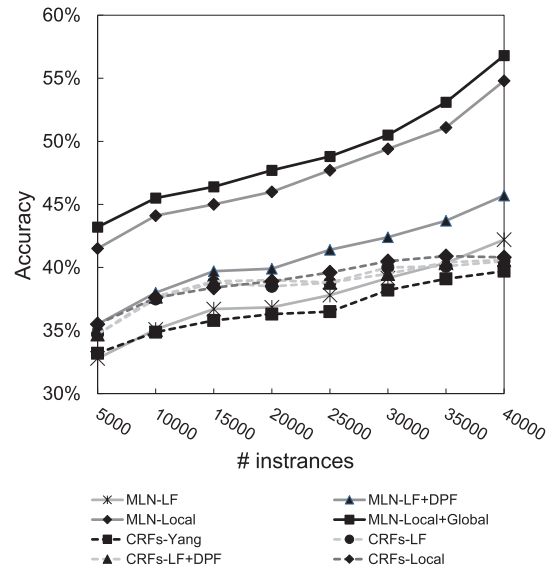


Figure 1: The impacts of training data size.

From the training procedures, we also empirically find that the training iterations of MLNs are small. It means that the convergence rate

of MLNs is fast. To evaluate the convergence rate, we also evaluate the dependence of the performances of MLNs on the number of training epochs. Figure 2 shows the results of *MLN-Local* and *MLN-Local+Global*. From the results, we can observe that the best performances can be achieved when the number of training epochs is more than nine. Hence, in this work, we set the number of iterations to be 10.

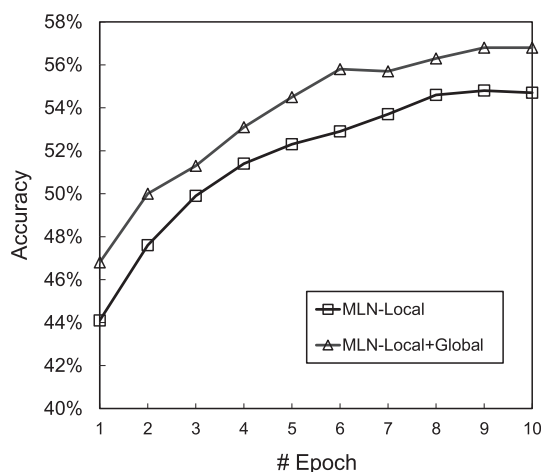


Figure 2: The performance curves on the number of training epochs.

## 5 Conclusions

In this paper, we focus on named entity abbreviation generation problem. We propose to use first-order logic to model rich linguistic features and global constraints. We convert the abbreviation generation to character deletion and keep operations. Linguistic features and relations between different operations are represented by local and global logic formulas respectively. Markov Logic Network frameworks is adopted for learning and predication. To reduce the human effort in constructing the training data, we also introduce an automatical training data construction methods with sample strategies. We collect standard forms of entities from online encyclopedia, use a few simple patterns to extract abbreviations from documents and search engine snippets with high precision as training data. Experimental results show that the proposed methods achieve better performance than state-of-the-art methods and can efficiently process large volumes of data.

## 6 Acknowledgement

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (61003092, 61073069), National Major Science and Technology Special Project of China (2014ZX03006005), Shanghai Municipal Science and Technology Commission (No.12511504502) and “Chen Guang” project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation(11CG05).

## References

- David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. 2011. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two, IJCAI'11*, pages 1171–1177. AAAI Press.
- AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40, Sydney, Australia, July. Association for Computational Linguistics.
- Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8.
- Jing-shin Chang and Yu-Tso Lai. 2004. A preliminary study on probabilistic models for chinese abbreviations. In *Proceedings of the Third SIGHAN Workshop on Chinese Language Learning*, pages 9–16.
- Jing-Shin Chang and Wei-Lun Teng. 2006. Mining atomic chinese abbreviations with a probabilistic single character recovery model. *Language Resources and Evaluation*, 40(3-4):367–374.
- Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Mkn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012. 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, EMNLP-CoNLL '12*, pages

- 421–432, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Minlie Huang, Xing Shi, Feng Jin, and Xiaoyan Zhu. 2012. Using first-order logic to compress sentences. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- 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, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Shangpu Jiang, D. Lowd, and Dejing Dou. 2012. Learning to refine an automatically extracted knowledge base using markov logic. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 912–917.
- Zhifei Li and David Yarowsky. 2008a. Mining and modeling relations between formal and informal Chinese phrases from web corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1031–1040, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Zhifei Li and David Yarowsky. 2008b. Unsupervised translation induction for chinese abbreviations using monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 425–433, Columbus, Ohio, June. Association for Computational Linguistics.
- Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. 2010. Pet: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 929–938, New York, NY, USA. ACM.
- Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 1035–1044, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. 2012. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 379–387, New York, NY, USA. ACM.
- Naoaki Okazaki, Mitsuru Ishizuka, and Jun'ichi Tsujii. 2008. A discriminative approach to japanese abbreviation extraction. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 889–894.
- Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, pages 650–659, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.
- Sebastian Riedel and Ivan Meza-Ruiz. 2008. Collective semantic role labelling with markov logic. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning, CoNLL '08*, pages 193–197, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA. ACM.
- P. Singla and P. Domingos. 2006. Entity resolution with markov logic. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 572–582.
- Li-Xing Xie, Ya-Bin Zheng, Zhi-Yuan Liu, Mao-Song Sun, and Can-Hui Wang. 2011. Extracting chinese abbreviation-definition pairs from anchor texts. In *Machine Learning and Cybernetics (ICMLC)*, volume 4, pages 1485–1491.
- Dong Yang, Yi-cheng Pan, and Sadaoki Furui. 2009. Automatic chinese abbreviation generation using conditional random field. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 273–276, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Dong Yang, Yi-Cheng Pan, and Sadaoki Furui. 2012. Vocabulary expansion through automatic abbreviation generation for chinese voice search. *Computer Speech & Language*, 26(5):321 – 335.
- Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with markov logic. 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 - Volume 1*, ACL '09, pages 405–413, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Full-coverage Identification of English Light Verb Constructions

István Nagy T.<sup>1</sup>, Veronika Vincze<sup>1,2</sup> and Richárd Farkas<sup>1</sup>

<sup>1</sup>Department of Informatics, University of Szeged  
{nistvan, rfarkas}@inf.u-szeged.hu

<sup>2</sup>Hungarian Academy of Sciences, Research Group on Artificial Intelligence  
vinczev@inf.u-szeged.hu

## Abstract

The identification of light verb constructions (LVC) is an important task for several applications. Previous studies focused on some limited set of light verb constructions. Here, we address the full coverage of LVCs. We investigate the performance of different candidate extraction methods on two English full-coverage LVC annotated corpora, where we found that less severe candidate extraction methods should be applied. Then we follow a machine learning approach that makes use of an extended and rich feature set to select LVCs among extracted candidates.

## 1 Introduction

A multiword expression (MWE) is a lexical unit that consists of more than one orthographical word, i.e. a lexical unit that contains spaces and displays lexical, syntactic, semantic, pragmatic and/or statistical idiosyncrasy (Sag et al., 2002; Calzolari et al., 2002). Light verb constructions (LVCs) (e.g. *to take a decision*, *to take sg into consideration*) form a subtype of MWEs, namely, they consist of a nominal and a verbal component where the verb functions as the syntactic head (the whole construction fulfills the role of a verb in the clause), but the semantic head is the noun (i.e. the noun is used in one of its original senses). The verbal component (also called a light verb) usually loses its original sense to some extent.<sup>1</sup> The meaning of LVCs can only partially be computed on the basis of the meanings of their parts and the way they are related to each other (semi-compositionality). Thus, the result of translating their parts literally can hardly be considered as

<sup>1</sup>*Light verbs* may also be defined as semantically empty support verbs, which share their arguments with a noun (see the NomBank project (Meyers et al., 2004)), that is, the term *support verb* is a hypernym of *light verb*.

the proper translation of the original expression. Moreover, the same syntactic pattern may belong to a LVC (e.g. *make a mistake*), a literal verb + noun combination (e.g. *make a cake*) or an idiom (e.g. *make a meal (of something)*), which suggests that their identification cannot be based on solely syntactic patterns. Since the syntactic and the semantic head of the construction are not the same, they require special treatment when parsing. On the other hand, the same construction may function as an LVC in certain contexts while it is just a productive construction in other ones, compare *He gave her a ring made of gold* (non-LVC) and *He gave her a ring because he wanted to hear her voice* (LVC).

In several natural language processing (NLP) applications like information extraction and retrieval, terminology extraction and machine translation, it is important to identify LVCs in context. For example, in machine translation we must know that LVCs form one semantic unit, hence their parts should not be translated separately. For this, LVCs should be identified first in the text to be translated.

As we shall show in Section 2, there has been a considerable amount of previous work on LVC detection, but some authors seek to capture just verb-object pairs, while others just verbs with prepositional complements. Actually, many of them exploited only constructions formed with a limited set of light verbs and identified or extracted just a specific type of LVCs. However, we cannot see any benefit that any NLP application could get from these limitations and here, we focus on the full-coverage identification of LVCs. We train and evaluate statistical models on the Wiki50 (Vincze et al., 2011) and Szeged-ParalellFX (SZPFX) (Vincze, 2012) corpora that have recently been published with full-coverage LVC annotation.

We employ a two-stage procedure. First,

we identify potential LVC candidates in running texts – we empirically compare various candidate extraction methods –, then we use a machine learning-based classifier that exploits a rich feature set to select LVCs from the candidates.

The main contributions of this paper can be summarized as follows:

- We introduce and evaluate systems for **identifying all LVCs and all individual LVC occurrences** in a running text and we do not restrict ourselves to certain specific types of LVCs.
- We systematically **compare and evaluate different candidate extraction methods** (earlier published methods and new solutions implemented by us).
- We defined and evaluated several **new feature templates** like semantic or morphological features to select LVCs in context from extracted candidates.

## 2 Related Work

Two approaches have been introduced for LVC detection. In the first approach, LVC candidates (usually verb-object pairs including one verb from a well-defined set of 3-10 verbs) are extracted from the corpora and these tokens – without contextual information – are then classified as LVCs or not (Stevenson et al., 2004; Tan et al., 2006; Fazly and Stevenson, 2007; Van de Cruys and Moirón, 2007; Gurrutxaga and Alegria, 2011). As a gold standard, lists collected from dictionaries or other annotated corpora are used: if the extracted candidate is classified as an LVC and can be found on the list, it is a true positive, regardless of the fact whether it was a genuine LVC in its context.

In the second approach, the goal is to detect individual LVC token instances in a running text, taking contextual information into account (Diab and Bhutada, 2009; Tu and Roth, 2011; Nagy T. et al., 2011). While the first approach assumes that a specific candidate in all of its occurrences constitutes an LVC or not (i.e. there are no ambiguous cases), the second one may account for the fact that there are contexts where a given candidate functions as an LVC whereas in other contexts it does not, recall the example of *give a ring* in Section 1.

The authors of Stevenson et al. (2004), Fazly and Stevenson (2007), Van de Cruys and Moirón

(2007) and Gurrutxaga and Alegria (2011) built LVC detection systems with statistical features. Stevenson et al. (2004) focused on classifying LVC candidates containing the verbs *make* and *take*. Fazly and Stevenson (2007) used linguistically motivated statistical measures to distinguish subtypes of verb + noun combinations. However, it is a challenging task to identify rare LVCs in corpus data with statistical-based approaches, since 87% of LVCs occur less than 3 times in the two full-coverage LVC annotated corpora used for evaluation (see Section 3).

A semantic-based method was described in Van de Cruys and Moirón (2007) for identifying verb-preposition-noun combinations in Dutch. Their method relies on selectional preferences for both the noun and the verb. Idiomatic and light verb noun + verb combinations were extracted from Basque texts by employing statistical methods (Gurrutxaga and Alegria, 2011). Diab and Bhutada (2009) and Nagy T. et al. (2011) employed ruled-based methods to detect LVCs, which are usually based on (shallow) linguistic information, while the domain specificity of the problem was highlighted in Nagy T. et al. (2011).

Both statistical and linguistic information were applied by the hybrid LVC systems (Tan et al., 2006; Tu and Roth, 2011; Samardžić and Merlo, 2010), which resulted in better recall scores. English and German LVCs were analysed in parallel corpora: the authors of Samardžić and Merlo (2010) focus on their manual and automatic alignment. They found that linguistic features (e.g. the degree of compositionality) and the frequency of the construction both have an impact on the alignment of the constructions.

Tan et al. (2006) applied machine learning techniques to extract LVCs. They combined statistical and linguistic features, and trained a random forest classifier to separate LVC candidates. Tu and Roth (2011) applied Support Vector Machines to classify verb + noun object pairs on their balanced dataset as candidates for true LVCs<sup>2</sup> or not. They compared the contextual and statistical features and found that local contextual features performed better on ambiguous examples.

---

<sup>2</sup>In theoretical linguistics, two types of LVCs are distinguished (Kearns, 2002). In true LVCs such as *to have a laugh* we can find a noun that is a converse of a verb (i.e. it can be used as a verb without any morphological change), while in vague action verbs such as *to make an agreement* there is a noun derived from a verb (i.e. there is morphological change).

Some of the earlier studies aimed at identifying or extracting only a restricted set of LVCs. Most of them focus on verb-object pairs when identifying LVCs (Stevenson et al., 2004; Tan et al., 2006; Fazly and Stevenson, 2007; Cook et al., 2007; Bannard, 2007; Tu and Roth, 2011), thus they concentrate on structures like *give a decision* or *take control*. With languages other than English, authors often select verb + prepositional object pairs (instead of verb-object pairs) and categorise them as LVCs or not. See, e.g. Van de Cruys and Moirón (2007) for Dutch LVC detection or Krenn (2008) for German LVC detection. In other cases, only true LVCs were considered (Stevenson et al., 2004; Tu and Roth, 2011). In some other studies (Cook et al., 2007; Diab and Bhutada, 2009) the authors just distinguished between the literal and idiomatic uses of verb + noun combinations and LVCs were classified into these two categories as well.

In contrast to previous works, we seek to identify all LVCs in running texts and do not restrict ourselves to certain types of LVCs. For this reason, we experiment with different candidate extraction methods and we present a machine learning-based approach to select LVCs among candidates.

### 3 Datasets

In our experiments, three freely available corpora were used. Two of them had fully-covered LVC sets manually annotated by professional linguists. The annotation guidelines did not contain any restrictions on the inner syntactic structure of the construction and both true LVCs and vague action verbs were annotated. The Wiki50 (Vincze et al., 2011) contains 50 English Wikipedia articles that were annotated for different types of MWEs (including LVCs) and Named Entities. SZPFX (Vincze, 2012) is an English–Hungarian parallel corpus, in which LVCs are annotated in both languages. It contains texts taken from several domains like fiction, language books and magazines. Here, the English part of the corpus was used.

In order to compare the performance of our system with others, we also used the dataset of Tu and Roth (2011), which contains 2,162 sentences taken from different parts of the British National Corpus. They only focused on true LVCs in this dataset, and only the verb-object pairs (1,039 positive and 1,123 negative examples) formed with the

verbs *do*, *get*, *give*, *have*, *make*, *take* were marked. Statistical data on the three corpora are listed in Table 1.

Corpus	Sent.	Tokens	LVCs	LVC lemma
Wiki50	4,350	114,570	368	287
SZPFX	14,262	298,948	1,371	706
Tu&Roth	2,162	65,060	1,039	430

Table 1: Statistical data on LVCs in the Wiki50 and SZPFX corpora and the Tu&Roth dataset.

Despite the fact that English verb + prepositional constructions were mostly neglected in previous research, both corpora contain several examples of such structures, e.g. *take into consideration* or *come into contact*, the ratio of such LVC lemmas being 11.8% and 9.6% in the Wiki50 and SZPFX corpora, respectively. In addition to the verb + object or verb + prepositional object constructions, there are several other syntactic constructions in which LVCs can occur due to their syntactic flexibility. For instance, the nominal component can become the subject in a passive sentence (*the photo has been taken*), or it can be extended by a relative clause (*the photo that has been taken*). These cases are responsible for 7.6% and 19.4% of the LVC occurrences in the Wiki50 and SZPFX corpora, respectively. These types cannot be identified when only verb + object pairs are used for LVC candidate selection.

Some researchers filtered LVC candidates by selecting only certain verbs that may be part of the construction, e.g. Tu and Roth (2011). As the full-coverage annotated corpora were available, we were able to check what percentage of LVCs could be covered with this selection. The six verbs used by Tu and Roth (2011) are responsible for about 49% and 63% of all LVCs in the Wiki50 and the SZPFX corpora, respectively. Furthermore, 62 different light verbs occurred in the Wiki50 and 102 in the SZPFX corpora, respectively. All this indicates that focusing on a reduced set of light verbs will lead to the exclusion of a considerable number of LVCs in free texts.

Some papers focus only on the identification of true LVCs, neglecting vague action verbs (Stevenson et al., 2004; Tu and Roth, 2011). However, we cannot see any NLP application that can benefit if such a distinction is made since vague action verbs and true LVCs share those properties that are relevant for natural language processing (e.g. they must be treated as one complex predicate (Vincze,



2012)). We also argue that it is important to separate LVCs and idioms because LVCs are semi-productive and semi-compositional – which may be exploited in applications like machine translation or information extraction – in contrast to idioms, which have neither feature. All in all, we seek to identify all verbal LVCs (not including idioms) in our study and do not restrict ourselves to certain specific types of LVCs.

## 4 LVC Detection

Our goal is to identify each LVC occurrence in running texts, i.e. to take input sentences such as *'We often have lunch in this restaurant'* and mark each LVC in it. Our basic approach is to syntactically parse each sentence and extract potential LVCs with different candidate extraction methods. Afterwards, a binary classification can be used to automatically classify potential LVCs as LVCs or not. For the automatic classification of candidate LVCs, we implemented a machine learning approach, which is based on a rich feature set.

### 4.1 Candidate Extraction

As we had two full-coverage LVC annotated corpora where each type and individual occurrence of a LVC was marked in running texts, we were able to examine the characteristics of LVCs in a running text, and evaluate and compare the different candidate extraction methods. When we examined the previously used methods, which just treated the verb-object pairs as potential LVCs, it was revealed that only 73.91% of annotated LVCs on the Wiki50 and 70.61% on the SZPFX had a verb-object syntactic relation. Table 2 shows the distribution of dependency label types provided by the Bohnet parser (Bohnet, 2010) for the Wiki50 and Stanford (Klein and Manning, 2003) and the Bohnet parsers for the SZPFX corpora. In order to compare the efficiency of the parsers, both were applied using the same dependency representation. In this phase, we found that the Bohnet parser was more successful on the SZPFX corpora, i.e. it could cover more LVCs, hence we applied the Bohnet parser in our further experiments.

We define the extended **syntax-based candidate extraction** method, where besides the *verb-direct object* dependency relation, the *verb-prepositional*, *verb-relative clause*, *noun-participial modifier* and *verb-subject of a passive construction* syntactic relations were also investi-

gated among verbs and nouns. Here, 90.76% of LVCs in the Wiki50 and 87.75% in the SZPFX corpus could be identified with the extended syntax-based candidate extraction method.

It should be added that some rare examples of split LVCs where the nominal component is part of the object, preceded by a quantifying expression like *he **gained much of his fame*** can hardly be identified by syntax-based methods since there is no direct link between the verb and the noun. In other cases, the omission of LVCs from candidates is due to the rare and atypical syntactic relation between the noun and the verb (e.g. *dep* in *reach conform*). Despite this, such cases are also included in the training and evaluation datasets as positive examples.

Edge type	Wiki50		SZPFX			
			Stanford		Bohnet	
dobj	272	73.91	901	65.71	968	70.6
pobj	43	11.69	93	6.78	93	6.78
nsubjpass	6	1.63	61	4.45	73	5.32
rmod	6	1.63	30	2.19	38	2.77
partmod	7	1.9	21	1.53	31	2.26
sum	334	90.76	1,106	80.67	1,203	87.75
other	15	4.07	8	0.58	31	2.26
none	19	5.17	257	18.75	137	9.99
sum	368	100.0	1,371	100.0	1,371	100.0

Table 2: Edge types in the Wiki50 and SZPFX corpora. dobj: object. pobj: preposition. nsubjpass: subject of a passive construction. rmod: relative clause. partmod: participial modifier. other: other dependency labels. none: no direct syntactic connection between the verb and noun.

Our second candidate extractor is the **morphology-based candidate extraction** method (Nagy T. et al., 2011), which was also applied for extracting potential LVCs. In this case, a token sequence was treated as a potential LVC if the POS-tag sequence matched one pattern typical of LVCs (e.g. VERB-NOUN). Although this method was less effective than the extended syntax-based approach, when we **merged the extended syntax-based and morphology-based methods**, we were able to identify most of the LVCs in the two corpora.

The authors of Stevenson et al. (2004) and Tu and Roth (2011) filtered LVC candidates by selecting only certain verbs that could be part of the construction, so we checked what percentage of LVCs could be covered with this selection when we treated just the verb-object pairs as LVC candidates. We found that even the least stringent selec-

tion covered only 41.88% of the LVCs in Wiki50 and 47.84% in SZPFX. Hence, we decided to drop any such constraint.

Table 3 shows the results we obtained by applying the different candidate extraction methods on the Wiki50 and SZPFX corpora.

Method	Wiki50		SZPFX	
	#	%	#	%
Stevenson et al. (2004)	107	29.07	372	27.13
Tu&Roth (2011)	154	41.84	656	47.84
doj	272	73.91	968	70.6
POS	293	79.61	907	66.15
Syntactic	334	90.76	1,203	87.75
POS $\cup$ Syntactic	339	92.11	1,223	89.2

Table 3: The recall of candidate extraction approaches. *doj*: verb-object pairs. *POS*: morphology-based method. *Syntactic*: extended syntax-based method. *POS  $\cup$  Syntactic*: union of the morphology- and extended syntax-based candidate extraction methods.

## 4.2 Machine Learning Based Candidate Classification

For the automatic classification of the candidate LVCs we implemented a machine learning approach, which we will elaborate upon below. Our method is based on a rich feature set with the following categories: statistical, lexical, morphological, syntactic, orthographic and semantic.

**Statistical features: Potential LVCs** were collected from 10,000 Wikipedia pages by the union of the morphology-based candidate extraction and the extended syntax-based candidate extraction methods. The number of their **occurrences** was used as a feature in case the candidate was one of the syntactic phrases collected.

**Lexical features:** We exploit the fact that the **most common verbs** are typically light verbs, so we selected fifteen typical light verbs from the list of the most frequent verbs taken from the corpora. In this case, we investigated whether the lemmatised verbal component of the candidate was one of these fifteen verbs. The **lemma of the head of the noun** was also applied as a lexical feature. The nouns found in LVCs were collected from the corpora, and for each corpus the noun list got from the union of the other two corpora was used. Moreover, we constructed **lists of lemmatised LVCs** from the corpora and for each corpus, the list got from the union of the other two corpora was utilised. In the case of the Tu&Roth dataset, the list got from Wiki50 and SZPFX was

filtered for the six light verbs and true LVCs they contained.

**Morphological features:** The POS candidate extraction method was used as a feature, so when the POS-tag sequence in the text matched one typical **‘POS-pattern’** of LVCs, the candidate was marked as *true*; otherwise as *false*. The **‘Verbal-Stem’** binary feature focuses on the stem of the noun. For LVCs, the nominal component is typically one that is derived from a verbal stem (*make a decision*) or coincides with a verb (*have a walk*). In this case, the phrases were marked as *true* if the stem of the nominal component had a verbal nature, i.e. it coincided with a stem of a verb. *Do* and *have* are often light verbs, but these verbs may occur as auxiliary verbs too. Hence we defined a feature for the two verbs to denote whether or not they were **auxiliary verbs** in a given sentence.

**Syntactic features:** The **dependency label** between the noun and the verb can also be exploited in identifying LVCs. As we typically found in the candidate extraction, the syntactic relation between the verb and the nominal component in an LVC is *doj*, *pobj*, *rcmod*, *partmod* or *nsubjpass* – using the Bohnet parser (Bohnet, 2010), hence these relations were defined as features. The **determiner** within all candidate LVCs was also encoded as another syntactic feature.

**Orthographic features:** in the case of the **‘suffix’** feature, it was checked whether the lemma of the noun ended in a given character bi- or trigram. It exploits the fact that many nominal components in LVCs are derived from verbs. The **‘number of words’** of the candidate LVC was also noted and applied as a feature.

**Semantic features:** In this case we also exploited the fact that the nominal component is derived from verbs. *Activity* or *event* semantic senses were looked for among the hypernyms of the noun in WordNet (Fellbaum, 1998).

We experimented with several learning algorithms and our preliminary results showed that decision trees performed the best. This is probably due to the fact that our feature set consists of a few compact – i.e. high-level – features. We trained the J48 classifier of the WEKA package (Hall et al., 2009), which implements the decision trees algorithm C4.5 (Quinlan, 1993) with the above-mentioned feature set. We report results with Support Vector Machines (SVM) (Cortes and Vapnik, 1995) as well, to compare our methods with Tu &

Method	Wiki50						SZPFX					
	J48			SVM			J48			SVM		
	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
DM	56.11	36.26	44.05	56.11	36.26	44.05	72.65	27.83	40.24	72.65	27.83	40.24
POS	60.65	46.2	52.45	54.1	48.64	51.23	66.12	43.02	52.12	54.88	42.42	47.85
Syntax	61.29	47.55	53.55	50.99	51.63	51.31	63.25	56.17	59.5	54.38	54.03	54.2
POS∪Syntax	58.99	51.09	54.76	49.72	51.36	50.52	63.29	56.91	59.93	55.84	55.14	55.49

Table 4: Results obtained in terms of precision, recall and F-score. DM: dictionary matching. POS: morphology-based candidate extraction. Syntax: extended syntax-based candidate extraction. POS ∪ Syntax: the merged set of the morphology-based and syntax-based candidate extraction methods.

Roth.

As the investigated corpora were not sufficiently big for splitting them into training and test sets of appropriate size, besides, the different annotation principles ruled out the possibility of enlarging the training sets with another corpus, we evaluated our models in 10-fold cross validation manner on the Wiki50, SZPFX and Tu&Roth datasets. But, in the case of Wiki50 and SZPFX, where only the positive LVCs were annotated, we employed  $F_{\beta=1}$  scores interpreted on the positive class as an evaluation metric. Moreover, we treated all potential LVCs as negative which were extracted by different extraction methods but were not marked as positive in the gold standard. The resulting datasets were not balanced and the number of negative examples basically depended on the candidate extraction method applied.

However, some positive elements in the corpora were not covered in the candidate classification step, since the candidate extraction methods applied could not detect all LVCs in the corpus data. Hence, we treated the omitted LVCs as false negatives in our evaluation.

## 5 Experiments and Results

As a baseline, we applied a context-free dictionary matching method. First, we gathered the gold-standard LVC lemmas from the two other corpora. Then we marked candidates of the union of the extended syntax-based and morphology-based methods as LVC if the candidate light verb and one of its syntactic dependents was found on the list.

Table 4 lists the results got on the Wiki50 and SZPFX corpora by using the baseline dictionary matching and our machine learning approach with different machine learning algorithm and different candidate extraction methods. The dictionary matching approach got the highest precision on SZPFX, namely 72.65%. Our machine learning-based approach with different candidate extraction

methods demonstrated a consistent performance (i.e. an F-score over 50) on the Wiki50 and SZPFX corpora. It is also seen that our machine learning approach with the union of the morphology- and extended syntax-based candidate extraction methods is the most successful method in the case of Wiki50 and SZPFX. On both corpora, it achieved an F-score that was higher than that of the dictionary matching approach (the difference being 10 and 19 percentage points in the case of Wiki50 and SZPFX, respectively).

In order to compare the performance of our system with others, we evaluated it on the Tu&Roth dataset (Tu and Roth, 2011) too. Table 5 shows the results got using dictionary matching, applying our machine learning-based approach with a rich feature set, and the results published in Tu and Roth (2011) on the Tu&Roth dataset. In this case, the dictionary matching method performed the worst and achieved an accuracy score of 61.25. The results published in Tu and Roth (2011) are good on the positive class with an F-score of 75.36 but the worst with an F-score of 56.41 on the negative class. Therefore this approach achieved an accuracy score that was 7.27 higher than that of the dictionary matching method. Our approach demonstrates a consistent performance (with an F-score over 70) on the positive and negative classes. It is also seen that our approach is the most successful in the case of the Tu&Roth dataset: it achieved an accuracy score of 72.51%, which is 3.99% higher than that got by the Tu&Roth method (Tu and Roth, 2011) (68.52%).

Method	Accuracy	F1+	F1-
DM	61.25	56.96	64.76
Tu&Roth Original	68.52	<b>75.36</b>	56.41
J48	<b>72.51</b>	74.73	<b>70.5</b>

Table 5: Results of applying different methods on the Tu&Roth dataset. DM: dictionary matching. Tu&Roth Original: the results of Tu and Roth (2011). J48: our model.

## 6 Discussion

The applied machine learning-based method extensively outperformed our dictionary matching baseline model, which underlines the fact that our approach can be suitably applied to LVC detection. As Table 4 shows, our presented method proved to be the most robust as it could obtain roughly the same recall, precision and F-score on the Wiki50 and SZPFX corpora. Our system’s performance primarily depends on the applied candidate extraction method. In the case of dictionary matching, a higher recall score was primarily limited by the size of the dictionary, but this method managed to achieve a fairly good precision score.

As Table 5 indicates, the dictionary matching method was less effective on the Tu&Roth dataset. Since the corpus was created by collecting sentences that contain verb-object pairs with specific verbs, this dataset contains a lot of negative and ambiguous examples besides annotated LVCs, hence the distribution of LVCs in the Tu&Roth dataset is not comparable to those in Wiki50 or SZPFX. In this dataset, only one positive or negative example was annotated in each sentence, and they examined just the verb-object pairs formed with the six verbs as a potential LVC. However, the corpus probably contains other LVCs which were not annotated. For example, in the sentence *it have been held that a gift to a charity of shares in a close company gave rise to a charge to capital transfer tax where the company had an interest in possession in a trust*, the phrase *give rise* was listed as a negative example in the Tu&Roth dataset, but *have an interest*, which is another LVC, was not marked either positive or negative. This is problematic if we would like to evaluate our candidate extractor on this dataset since it would identify this phrase, even if it is restricted to verb-object pairs containing one of the six verbs mentioned above, thus yielding false positives already in the candidate extraction phase.

Moreover, the results got with our machine learning approach overperformed those reported in Tu and Roth (2011). This may be attributed to the inclusion of a rich feature set with new features like semantic or morphological features that was used in our system, which demonstrated a consistent performance on the positive and negative classes too.

To examine the effectiveness of each individual feature of the machine learning based candidate

classification, we carried out an ablation analysis. Table 6 shows the usefulness of each individual feature type on the SZPFX corpus.

Feature	Precision	Recall	F-score	Diff
Statistical	60.55	55.88	58.12	-1.81
Lexical	71.28	28.6	40.82	-19.11
Morphological	62.3	54.77	58.29	-1.64
Syntactic	59.87	55.8	57.77	-2.16
Semantic	60.81	54.77	57.63	-2.3
Orthographic	63.3	56.25	59.56	-0.37
All	<b>63.29</b>	<b>56.91</b>	<b>59.93</b>	-

Table 6: The usefulness of individual features in terms of precision, recall and F-score using the SZPFX corpus.

For each feature type, we trained a J48 classifier with all of the features except that one. We then compared the performance to that got with all the features. As our ablation analysis shows, each type of feature contributed to the overall performance. The most important feature is the list of the most frequent light verbs. The most common verbs in a language are used very frequently in different contexts, with several argument structures and this may lead to the bleaching (or at least generalization) of its semantic content (Altmann, 2005). From this perspective, it is linguistically plausible that the most frequent verbs in a language largely coincide with the most typical light verbs since light verbs lose their original meaning to some extent (see e.g. Sanromán Vilas (2009)).

Besides the ablation analysis we also investigated the decision tree model yielded by our experiments. Similar to the results of our ablation analysis we found that the lexical features were the most powerful, the semantic, syntactic and orthographical features were also useful while statistical and morphological features were less effective but were still exploited by the model.

Comparing the results on the three corpora, it is salient that the F-score got from applying the methods on the Tu&Roth dataset was considerably better than those got on the other two corpora. This can be explained if we recall that this dataset applies a restricted definition of LVCs, works with only verb-object pairs and, furthermore, it contains constructions with only six light verbs. However, Wiki50 and SZPFX contain all LVCs, they include verb + preposition + noun combinations as well, and they are not restricted to six verbs. All these characteristics demonstrate that identifying LVCs in the latter two corpora is a more realistic

and challenging task than identifying them in the artificial Tu&Roth dataset. For example, the very frequent and important LVCs like *make a decision*, which was one of the most frequent LVCs in the two full-coverage LVC annotated corpora, are ignored if we only focus on identifying true LVCs. It could be detrimental when a higher level NLP application exploits the LVC detector.

We also carried out a manual error analysis on the data. We found that in the candidate extraction step, it is primarily POS-tagging or parsing errors that result in the omission of certain LVC candidates. In other cases, the dependency relation between the nominal and verbal component is missing (recall the example of objects with quantifiers) or it is an atypical one (e.g. *dep*) not included in our list. The lower recall in the case of SZPFX can be attributed to the fact that this corpus contains more instances of nominal occurrences of LVCs (e.g. *decision-making* or *record holder*) than Wiki50, which were annotated in the corpora but our morphology-based and extended syntax-based methods were not specifically trained for them since adding POS-patterns like NOUN-NOUN or the corresponding syntactic relations would have resulted in the unnecessary inclusion of many nominal compounds.

As for the errors made during classification, it seems that it was hard for the classifier to label longer constructions properly. It was especially true when the LVC occurred in a non-canonical form, as in a relative clause (*counterargument that can be made*). Constructions with atypical light verbs (e.g. *cast a glance*) were also somewhat more difficult to find. Nevertheless, some false positives were due to annotation errors in the corpora. A further source of errors was that some literal and productive structures like *to give a book (to someone)* – which contains one of the most typical light verbs and the noun is homonymous with the verb *book* “to reserve” – are very difficult to distinguish from LVCs and were in turn marked as LVCs. Moreover, the classification of idioms with a syntactic or morphological structure similar to typical LVCs – *to have a crush on someone* “to be fond of someone”, which consists of a typical light verb and a deverbal noun – was also not straightforward. In other cases, verb-particle combinations followed by a noun were labeled as LVCs such as *make up his mind* or *give in his notice*. Since Wiki50 contains annotated ex-

amples for both types of MWEs, the classification of verb + particle/preposition + noun combinations as verb-particle combinations, LVCs or simple verb + prepositional phrase combinations could be a possible direction for future work.

## 7 Conclusions

In this paper, we introduced a system that enables the full coverage identification of English LVCs in running texts. Our method detected a broader range of LVCs than previous studies which focused only on certain subtypes of LVCs. We solved the problem in a two-step approach. In the first step, we extracted potential LVCs from a running text and we applied a machine learning-based approach that made use of a rich feature set to classify extracted syntactic phrases in the second step. Moreover, we investigated the performance of different candidate extraction methods in the first step on the two available full-coverage LVC annotated corpora, and we found that owing to the overly strict candidate extraction methods applied, the majority of the LVCs were overlooked. Our results show that a full-coverage identification of LVCs is challenging, but our approach can achieve promising results. The tool can be used in preprocessing steps for e.g. information extraction applications or machine translation systems, where it is necessary to locate lexical items that require special treatment.

In the future, we would like to improve our system by conducting a detailed analysis of the effect of the features included. Later, we also plan to investigate how our LVC identification system helps higher level NLP applications. Moreover, we would like to adapt our system to identify other types of MWE and experiment with LVC detection in other languages as well.

## Acknowledgments

This work was supported in part by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013).

## References

Gabriel Altmann. 2005. Diversification processes. In *Handbook of Quantitative Linguistics*, pages 646–659, Berlin. de Gruyter.

- Colin Bannard. 2007. A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In *Proceedings of MWE 2007*, pages 1–8, Morristown, NJ, USA. ACL.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of Coling 2010*, pages 89–97.
- Nicoletta Calzolari, Charles Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, Catherine MacLeod, and Antonio Zampolli. 2002. Towards best practice for multiword expressions in computational lexicons. In *Proceedings of LREC 2002*, pages 1934–1940, Las Palmas.
- Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of MWE 2007*, pages 41–48, Morristown, NJ, USA. ACL.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.
- Mona Diab and Pravin Bhutada. 2009. Verb Noun Construction MWE Token Classification. In *Proceedings of MWE 2009*, pages 17–22, Singapore, August. ACL.
- Afsaneh Fazly and Suzanne Stevenson. 2007. Distinguishing Subtypes of Multiword Expressions Using Linguistically-Motivated Statistical Measures. In *Proceedings of MWE 2007*, pages 9–16, Prague, Czech Republic, June. ACL.
- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May.
- Antton Gurrutxaga and Iñaki Alegria. 2011. Automatic Extraction of NV Expressions in Basque: Basic Issues on Cooccurrence Techniques. In *Proceedings of MWE 2011*, pages 2–7, Portland, Oregon, USA, June. ACL.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.
- Kate Kearns. 2002. *Light verbs in English*. Manuscript.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Annual Meeting of the ACL*, volume 41, pages 423–430.
- Brigitte Krenn. 2008. Description of Evaluation Resource – German PP-verb data. In *Proceedings of MWE 2008*, pages 7–10, Marrakech, Morocco, June.
- Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank Project: An Interim Report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA. ACL.
- István Nagy T., Veronika Vincze, and Gábor Berend. 2011. Domain-Dependent Identification of Multiword Expressions. In *Proceedings of the RANLP 2011*, pages 622–627, Hissar, Bulgaria, September. RANLP 2011 Organising Committee.
- Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of CICLing 2002*, pages 1–15, Mexico City, Mexico.
- Tanja Samardžić and Paola Merlo. 2010. Cross-lingual variation of light verb constructions: Using parallel corpora and automatic alignment for linguistic research. In *Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground*, pages 52–60, Uppsala, Sweden, July. ACL.
- Begoña Sanromán Vilas. 2009. Towards a semantically oriented selection of the values of Oper<sub>1</sub>. The case of *golpe* ‘blow’ in Spanish. In *Proceedings of MTT 2009*, pages 327–337, Montreal, Canada. Université de Montréal.
- Suzanne Stevenson, Afsaneh Fazly, and Ryan North. 2004. Statistical Measures of the Semi-Productivity of Light Verb Constructions. In *MWE 2004*, pages 1–8, Barcelona, Spain, July. ACL.
- Yee Fan Tan, Min-Yen Kan, and Hang Cui. 2006. Extending corpus-based identification of light verb constructions using a supervised learning framework. In *Proceedings of MWE 2006*, pages 49–56, Trento, Italy, April. ACL.
- Yuancheng Tu and Dan Roth. 2011. Learning English Light Verb Constructions: Contextual or Statistical. In *Proceedings of MWE 2011*, pages 31–39, Portland, Oregon, USA, June. ACL.
- Tim Van de Cruys and Begoña Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of MWE 2007*, pages 25–32, Morristown, NJ, USA. ACL.
- Veronika Vincze, István Nagy T., and Gábor Berend. 2011. Multiword Expressions and Named Entities in the Wiki50 Corpus. In *Proceedings of RANLP 2011*, pages 289–295, Hissar, Bulgaria, September. RANLP 2011 Organising Committee.
- Veronika Vincze. 2012. Light Verb Constructions in the SzegedParalellFX English–Hungarian Parallel Corpus. In *Proceedings of LREC 2012*, Istanbul, Turkey.

# Detecting deceptive opinions with profile compatibility

**Vanessa Wei Feng**

University of Toronto  
Toronto, ON, M5S 3G4, Canada  
weifeng@cs.toronto.edu

**Graeme Hirst**

University of Toronto  
Toronto, ON, M5S 3G4, Canada  
gh@cs.toronto.edu

## Abstract

We propose using profile compatibility to differentiate genuine and fake product reviews. For each product, a *collective* profile is derived from a separate collection of reviews. Such a profile contains a number of aspects of the product, together with their descriptions. For a given unseen review about the same product, we build a *test* profile using the same approach. We then perform a bidirectional alignment between the *test* and the *collective* profile, to compute a list of aspect-wise compatible features. We adopt Ott et al. (2011)'s *op\_spam\_v1.3* dataset for identifying *truthful* vs. *deceptive* reviews. We extend the recently proposed N-GRAM+SYN model of Feng et al. (2012a) by incorporating profile compatibility features, showing such an addition significantly improves upon their state-of-art classification performance.

## 1 Introduction

With the rapid development of e-commerce and the increasing popularity of various product review websites, people are more and more used to making purchase decisions based on the reported experience of other customers. A product rated positively by its previous users is able to attract potential new customers, while a poorly rated product is certainly not a good option for most new customers. Given this influential power of product reviews, there comes a huge potential for *deceptive opinion spam* to distort the true evaluation of a product. The promoters of a product may post false complimentary reviews, and competitors may post false derogatory reviews.

Although the task of detecting *deceptive opinion spam* can be formulated as a traditional bi-

nary classification problem with two classes, *deceptive* and *truthful*, where supervised learning can be applied, it is essentially very challenging. Its major difficulty arises from the lack of reliably labeled data, because it is extremely difficult for humans to identify through reading (see discussion in Section 4). Therefore, much previous work on detecting deceptive opinions usually relies on some meta-information, such as the IP address of the reviewer or the average rating of the product, rather than the actual content of the review Liu (2012). However, thanks to the release of the *op\_spam\_v1.3* dataset by Ott et al. (2011), a gold-standard dataset composed of 400 truthful and 400 deceptive hotel reviews (see Section 3), we are now at an appropriate stage for conducting supervised learning and reliable evaluation of the task.

In their work, Ott et al. proposed to use *n*-grams as features, and trained an SVM classifier to classify whether a given review is deceptive or truthful, achieving nearly 90% accuracy. More recently, Feng et al. (2012a) extended Ott et al.'s *n*-gram feature set by incorporating deep syntax features, i.e., syntactic production rules derived from Probabilistic Context Free Grammar (PCFG) parse trees, and obtained 91.2% accuracy on the same dataset.

While both Ott et al. and Feng et al.'s cues of deception come purely from the surface realization in the reviews, we propose the use of additional signals of truthfulness by characterizing the degree of *compatibility* between the personal experience described in a test review and a product profile derived from a collection of reference reviews about the same product. Our intuition comes from the hypothesis that since the writer of a deceptive review usually does not have any actual experience with that product, the resulting review might contain some contradictions with facts about the product, or the writer might fail to mention those as-

pects of the product that are commonly mentioned in truthful reviews. Conversely, a writer of a truthful review is more likely to make similar comments about some particular aspects of the product as other truthful reviewers. In other words, we postulate that by aligning a the profile of a product and the description of the writer’s personal experience, some useful clues can be revealed to help identify possible deception.

In line with the intuition above, we want to capture two types of compatibility: (1) Compatibility with the existence of some *distinct* aspect of the product, such as the mention of the famous art museum nearby the hotel; (2) Compatibility with the description of some *general* aspect of the product, such as commenting that the breakfast at the hotel is charged for.

We show that, by incorporating our computed profile alignment features with Feng et al.’s *n*-grams and deep syntax features, we significantly improve on the state-of-art performance presented by their N-GRAM+SYN model.

## 2 Related work

Before the existence of gold-standard datasets which include both truthful and deceptive product reviews, several attempts were made to detect deceptive opinions.

With regard to unsupervised approaches, previous work was primarily based on various patterns of atypical behaviors. Lim et al. (2010) proposed several behavior models to identify unusual reviewer patterns; e.g., spammers may contribute a fake review soon after product launch to maximally affect subsequent reviewers. Each of these individual models assigns a numeric score indicating the extent to which the reviewer has engaged in apparent spam behaviors, and these scores are then combined to product a final spam score. Jindal et al. (2010) employed data-mining techniques to discover unexpected class association rules; e.g., reviewers who give all high ratings to products of a brand while most other reviews are generally negative about the brand. Wu et al. (2010) proposed to use the distortion of popularity rankings as an indicator of opinion spamming, in the sense that deleting a set of random reviews should not overly disrupt the popularity ranking of a list of entities, while deleting fake reviews should significantly distort the overall rankings, under the assumption that deceptive reviews usually express a

sentiment at odds with legitimate reviews. Feng et al. (2012b) postulated that for a given domain, there exists a set of representative distributions of review rating scores, and fake reviews written by hired spammers will distort such natural distributions. Experimenting with a range of pseudo-gold-standard datasets, they provided quantitative insights into the characteristics of natural distributions of opinions in various product review domains.

With respect to supervised approaches, Jindal and Liu (2008) first conducted a tentative study of detecting deceptive opinions. In the absence of gold-standard datasets, they trained models using features from the review texts, as well as from meta-information on the reviewer and the product, to distinguish between *duplicate* reviews (regarded as deceptive), i.e., reviews whose major content appears more than once in the corpus, and *non-duplicate* reviews (regarded as truthful). However, these (near-)duplicate reviews are often not sophisticated in their composition, and therefore are relatively easy to identify, even by off-the-shelf plagiarism detection software (Ott et al., 2011).

Yoo and Gretzel (2009) constructed a small gold-standard dataset with 40 truthful and 42 deceptive hotel reviews, and manually inspected the statistical differences of psychologically relevant linguistic features for truthful and deceptive reviews.

Ott et al. (2011) released the *op\_spam\_v1.3* dataset (see Section 3), a much larger dataset with 400 truthful and 400 deceptive hotel reviews with *positive* comments, which allows the application of machine learning techniques for training models to automatically detect deceptive opinions. In their work, Ott et al. focused on the textual content of the reviews, and approached the task of detecting deceptive opinions by finding any *stretch of imagination* in the text — they treated fake reviews as imaginative writing, and used standard computational approaches of *genre identification*, *psycholinguistic deception detection*, and *text categorization* to identify them. Among the set of features explored in their classification experiments, they discovered that the features traditionally employed in either psychological studies of deception or genre identification were both significantly outperformed by a much simpler N-GRAM model with *n*-grams only as features, which achieved



nearly 90% accuracy. Later, Feng et al. (2012a) proposed a strengthened model, N-GRAM+SYN, by incorporating syntactic production rules derived from Probabilistic Context Free Grammar (PCFG) parse trees. They obtained 91.2% accuracy on the `op_spam_v1.3` dataset, which is a 14% error reduction.

More recently, Ott et al. released a new version, the `op_spam_v1.4` dataset, with gold-standard *negative* reviews included as well (Ott et al., 2013), which offers an opportunity to more extensively study the problem of detecting deceptive reviews. However, the work described in this paper is focused on *positive* review spam, and based directly on the work of Feng et al. We extend their N-GRAM+SYN model by incorporating potential signals of truthfulness, derived from aligning the description in a test review with the product profile constructed from a large collection of reference reviews about the same product. Somewhat orthogonal to Feng et al.’s *n*-grams and deep syntax features, which are focused on the surface realization of a given product review, our alignment features are able to employ useful information from the product itself.

### 3 The `op_spam_v1.3` Dataset

Due to the difficulty for humans to identify deceptive opinions, traditional approaches to constructing annotated corpora by recruiting human judges to label a given set of texts does not apply to the task of deception detection. Consequently, crowdsourcing services such as Amazon Mechanical Turk<sup>1</sup> (AMT) have been adopted as a better solution (Ott et al., 2011; Rubin and Vashchilko, 2012). By asking paid subjects on AMT to compose deceptive and/or truthful texts, corpora with reliable labels can be constructed.

In this work, we use Ott et al.’s `op_spam_v1.3` dataset<sup>2</sup>, which contains **positive** reviews for the 20 most-rated hotels on TripAdvisor<sup>3</sup> (TA) in Chicago. For each hotel, Ott et al. selected 20 deceptive reviews from submissions on AMT, and 20 truthful reviews from 5-star reviews on TA, resulting in 800 reviews in total for 20 hotels. The average length of deceptive reviews is 115.75 words, while the truthful reviews are chosen to have roughly the same average length.

<sup>1</sup><http://mturk.com>.

<sup>2</sup><http://www.cs.cornell.edu/~myleott>

<sup>3</sup><http://tripadvisor.com>.

## 4 Difficulty of the Task

It has been shown that humans can differentiate truthful reviews from deceptive ones with only a modest accuracy. With respect to the `op_spam_v1.3` dataset used in our work, human judges were only able to achieve 60% accuracy, and the inter-annotator agreement was low: 0.11 computed by Fleiss’s kappa (Ott et al., 2011). Since as humans, we do not have a tangible intuition of what signals deception and what characterizes truthfulness, we have great difficulty designing features for automatic identifying deceptive opinions.

In addition, there is difficulty with regard to the construction of gold-standard datasets for the task of deception detection. On one hand, although crowdsourcing provides a relatively cheap and reliable approach, gathering deceptive data is still nevertheless laborious. On the other hand, since for truthful reviews, we are able to select a small subset for which we have high confidence only by using a combination of various heuristics, the constructed dataset is inherently biased. Therefore, given the limited number and size of currently available gold-standard datasets, in the process of developing more sophisticated models involving a larger number of features, we may inevitably face the issue of overfitting on a relatively small set of data.

## 5 Methodology

### 5.1 Baseline features

We adopt Feng et al.’s N-GRAM+SYN model as the baseline system, which employs two distinct sets of features.

**N-GRAM features:** As shown by Ott et al., the bag-of-words model is effective for identifying deceptive reviews. However, the optimal choice of *n* is not consistent in Ott et al.’s original implementation and Feng et al.’s strengthened model: Ott et al. used the union of unigrams and bigrams, while Feng et al. obtained their best performance using unigrams alone, together with deep syntax features. Therefore, for a fair comparison, we consider using unigrams, bigrams, and the union of both, and choose the best combination with deep syntax features as our baseline system.

**SYN features:** Following Feng et al., deep syntax features are encoded as production rules derived from PCFG parse trees. These production

rules include lexicalized ones, i.e., POS-tag-to-word rules, and are combined with the grandparent node.

All baseline features are encoded as their tf-idf weights, with features appearing only once in the dataset eliminated.

## 5.2 Product profile construction

### 5.2.1 Aspects and compatibility

As introduced in Section 1, we postulate that by aligning the profile of a product with the description of the writer’s personal experience, we can characterize the degree of compatibility between them. Such alignment features could serve as useful signals to differentiate between deception and truthfulness.

In particular, we define compatibility for two types of aspects of a product — *distinct* aspects and *general* aspects. Distinct aspects are special features of the product. For hotels, those distinct aspects are usually realized as proper noun phrases in the text, typically the names of the landmarks nearby the hotel, including museums, parks, restaurants, bars, and shopping malls. On the other hand, general aspects are common features which typically appear in any product of this particular kind. For hotels, **location**, **service**, and **breakfast** are typical general aspects. The method we use to identify and extract distinct and general aspects from reviews is described in Section 5.2.4.

Compatibility for distinct and general aspects is defined as follows:

1. Compatibility with the **existence** of some *distinct* aspect of the product, e.g., truthful reviews of this Chicago hotel often mention the famous nearby Field Museum, and the test review also mentions this museum.
2. Compatibility with the **description** of some *general* aspect of the product. For example, breakfast at most Chicago hotels is complimentary; however, the breakfast at this hotel is charged for, and the test review describes it as “the breakfast is kinda expensive”.

### 5.2.2 Definition of product profile

A product’s profile  $P$  is composed of a set of its aspects  $A = \{a_1, \dots, a_N\}$ , where each  $a_i$  is an individual aspect of the product. Each aspect  $a_i$  is associated with a description  $D_i = \{p_1 : w_1, \dots, p_m : w_m\}$ , in which each element is a unique word  $p_j$

Aspect	Description
Bathroom	{ <i>clean</i> : 3.0, <i>comfortable</i> : 3.0, <i>pleasant</i> : 5.0, <i>high-end</i> : 1.0, <i>European</i> : 1.0}
Room	{ <i>wonderful</i> : 4.0, <i>deluxe</i> : 2.0, <i>huge</i> : 2.0}
Service	{ <i>average</i> : 2.0, <i>slow</i> : 2.0}
Michigan Ave.	{ <i>existence</i> : 5.0}

Table 1: An example fragment of the profile of a hotel.

to describe the aspect, along with the weight  $w_j$  assigned to that word.

Distinct aspects,  $a_{i,d}$ , and general aspects,  $a_{i,g}$ , are treated differently when it comes to their descriptions. For a distinct aspect, its description can contain only one word, *existence*, because we only care about whether this aspect is mentioned in the text, not the particular words used to describe this aspect. In fact, most of these distinct aspects do not occur with any associated adjectives or adverbs.

An example fragment of a hotel’s profile is shown in Table 1, in which the last aspect **Michigan Ave.** is a distinct aspect, and thus only one word, *existence*, appears in its description.

### 5.2.3 Data for profile construction

To establish a profile  $P$  for each target hotel in the op\_spam\_v1.3 dataset, we first gather all its reviews on TripAdvisor, from which we choose up to 200 reviews, subject to the following criteria:

1. It is not present in the op\_spam\_v1.3 dataset.
2. Its rating is five stars (the highest).
3. Its language is English.
4. It contains at least 150 characters.
5. The author has written at least 10 reviews.
6. The author has written reviews for at least 5 different hotels.
7. The author has received at least 5 helpfulness votes from other users.

If there are more than 200 reviews satisfying the above criteria, we choose the 200 with the highest number of helpfulness votes received by their authors. Most of the above criteria are consistent

<b>Reviews</b>			
	<i>Min</i>	<i>Max</i>	<i>Mean</i>
Reviews per hotel	44	200	160.6
Characters	150	8,995	848.3
<b>Authors</b>			
	<i>Min</i>	<i>Max</i>	<i>Mean</i>
Total reviews	10	506	37.8
Reviewed hotels	5	278	17.8
Helpfulness votes	5	2,280	33.3

Table 2: The statistics of the collection of reviews used in constructing profiles and their distinct authors.

with the original setup of collecting the truthful reviews in the `op_spam_v1.3` dataset. However, we add the last three strong criteria to ensure the quality of the collected reviews.

Table 2 lists the statistics of the reviews used in our profile construction. The statistics are categorized into (1) the information about the reviews themselves, including the total number of qualified reviews for each hotel, and the length (in characters) of each review; and (2) the information about the distinct authors of these reviews, including the total number of reviews each author has written on TripAdvisor, the total number of hotels each author has reviewed, and the total number of helpfulness votes that each author has received from other users.

#### 5.2.4 Aspect extraction

**Raw aspect extraction** For each particular hotel  $h$ , from the set of its corresponding reviews  $R_h$ , we extract the aspects  $A_h = \{a_{1,t_1,h}, \dots, a_{M,t_M,h}\}$  (e.g., **room**, **service**, and **Michigan Ave.** in Table 1) to be included into  $h$ 's profile  $P_h$ , where  $t_i$  is the type, *distinct* or *general*, of each aspect. For the sake of notational clarity, from now on, when talking about a specific hotel  $h$ , we will omit the subscript  $h$ . We first extract all noun phrases present in  $R$ , and use these noun phrases as raw aspects. For each raw aspect extracted, we identify whether it is a *distinct* or *general* aspect: if the aspect is extracted as a proper noun phrase (as tagged by the Stanford parser (Klein and Manning, 2003)), then the aspect is labeled as *distinct*, and *general* otherwise.

Noun phrase extraction is performed solely on

the syntactic parse trees. We do not attempt to resolve coreferences for the following two reasons: (1) since most coreference resolution tools are trained on news texts, their performance on product reviews might not be reliable; (2) we observe that authors of these product reviews rarely employ coreference in their writing, and therefore the impact of resolving coreference might be minor after all.

**Aspect clustering** Due to the flexibility of natural languages, it is common to see people express the same concept through different lexical usage. For example, both “price” and “rate” can refer to the price of the hotel. In addition to synonyms, semantically related words can also be adopted to describe the same aspect of the product, for example, “lighting” and “painting” can both describe the decoration of the hotel.

In order to deal with the use of these (near-) synonyms in text, we perform separate clustering for the sets of distinct and general aspects.

For distinct aspects  $\{a_{i,d}\}$ , the distance between a pair of aspects is defined by the length of their longest common substring divided by the average length of the two aspects. The motivation for this definition is to bring *distinct* aspects such as **Michigan Ave.** and **Michigan Avenue** into the same cluster.

For general aspects  $\{a_{i,g}\}$ , the distance between a pair of aspects is defined as their Lin Similarity (Lin, 1998), implemented by NLTK’s similarity package (Bird et al., 2009).

For both types of aspects, we perform hierarchical agglomerative clustering with average linkage. Cluster merging is terminated once the distance between all pairs of clusters is greater than 0.3.

**Sorting aspects by occurrences** After clustering distinct and general aspects separately, we keep track of the total occurrences of each aspect across the entire set  $R$ . Then, we let  $A$  be the set of the most common  $N$  aspects<sup>4</sup>.  $N$  is a parameter identical for all hotels, indicating how many aspects are included in each profile. We expect that the choice of  $N$  is affected by the particular type of product, and we experiment with several values of  $N$ .

<sup>4</sup> $N$  is usually much smaller than  $M$ , the total number of raw aspects.

### 5.2.5 Description extraction

For a hotel  $h$ , each of its aspects  $a_i$  is then associated with a description

$$D_i = \{p_{i,1} : w_{i,1}, \dots, p_{i,m} : w_{i,m}\},$$

which has two components: description words  $p_{i,j}$  and their weights  $w_{i,j}$ , for  $1 \leq j \leq m$ .

**Description words** The list  $\{p_{i,1}, \dots, p_{i,m}\}$  is composed of the description words used to modify the particular aspect  $a_i$ , e.g.,  $\{\text{wonderful}, \text{deluxe}, \text{and huge}\}$  for the aspect **room** in Table 1.

For each *distinct* aspect  $a_{i,d}$ , as discussed in Section 5.2.2, there can be only one description word, *existence*, indicating the presence of this aspect in the profile.

For each *general* aspect  $a_{i,g}$ , its description words are automatically extracted from the dependencies in the set of reviews  $R$ . From the dependencies output by the Stanford dependency parser (de Marneffe et al., 2006), for a particular aspect  $a_i$ , we first locate all relevant dependency relations which have  $a_i$  (or the head noun of  $a_i$  if  $a_i$  is a noun phrase) as the governors (or dependents, depending on the particular dependency types). Then from these relevant dependency relations, we extract the words that appear in the dependent (or governor) slots to be the description words, provided that those words are tagged as adjectives or adverbs.

**Weight assignment** Each description word  $p_{i,j}$  is assigned a weight  $w_{i,j}$  to indicate the *reliability* of that particular description word. There are many possibilities for these weight assignment functions. Here, as a preliminary study, for *distinct* aspects  $\{a_{i,d}\}$ , we simply use the number of occurrences of the aspect in the review collection  $R$  as the assigned weight  $w_{i,j}$ , and for *general* aspects  $\{a_{i,g}\}$ , we use the number of occurrences of  $p_{i,j}$  in  $R$  as  $w_{i,j}$ .

### 5.3 Computing profile compatibility

Once we establish a profile  $P$  for each hotel  $h$  from the set of relevant reviews  $R$ , then given an unseen test review  $r$  about  $h$ , we can perform a *bidirectional* alignment with  $P$ , and compute a list of compatibility features.

Using the same approach as described in Sections 5.2.4 and 5.2.5, we construct a profile  $Q$  from this single review. For the sake of clarity, we call  $Q$  the *test profile*, in contrast to  $P$ , the *collective*

*profile*, which is constructed from a collection of reviews.

#### 5.3.1 Bidirectional alignment

We perform a *bidirectional* alignment between the test profile  $P$  and the collective profile  $Q$ . The *direction* of an alignment is defined as the following.

When aligning the test profile  $Q$  with the collective profile  $P$  ( $Q \rightarrow P$ ), we compute *aspect-wise* compatibility features (to be defined in Section 5.3.2) **for each aspect  $a_i$  in  $P$** . Similarly, when aligning the collective profile  $P$  with the test profile  $Q$  ( $P \rightarrow Q$ ), we compute **for each aspect  $a_i$  in  $Q$** .

The  $Q \rightarrow P$  alignment captures whether the aspects that most reviewers would describe in their reviews are mentioned in the test review, and whether there exist any similarity or conflicts between the common opinions represented in the collective profile  $P$  and the test review.

The  $P \rightarrow Q$  alignment captures whether the test review falsely includes some imaginary extra aspects about the product, such as an indoor pool in the hotel, while the collective profile  $P$  does not include this aspect.

#### 5.3.2 Aspect-wise compatibility features

For each direction of the bidirectional alignment, we compute a list of aspect-wise compatibility features. Without loss of generality, assume that the direction of the alignment is  $Q \rightarrow P$ . Thus, for each aspect  $a_i$  in  $P$ , we include three features to indicate the compatibility between  $P$  and  $Q$ :

1.  $m_{a_i}$ : a boolean value indicating whether  $a_i$  is mentioned in  $Q$  or not. If  $a_i$  does appear in  $Q$ , we then compute the following two values, based on  $a_i$ 's description  $D_i$  in  $P$ , and its description  $D'_i$  in  $Q$ .
2.  $s_{a_i} \in [0, 1]$ : a numeric value indicating the *similarity* between  $P$  and  $Q$  with respect to  $a_i$ .  $S_{a_i}$  is computed as the cosine similarity between  $D_i$  and  $D'_i$ . We treat  $D_i$  and  $D'_i$  as two vectors following the bag-of-words model, where the description words are the words, and the weights are their corresponding frequencies.
3.  $c_{a_i} \in [0, 1]$ : a numeric value indicating the *conflicts* between  $P$  and  $Q$  with respect to  $a_i$ . For each description word  $p_i$  in  $D_i$ , we determine whether  $D'_i$  includes its antonym.

Antonyms are determined using the lexical relations defined in WordNet<sup>5</sup>.  $c_{a_i}$  is computed as the fraction of description words in  $D_i$  that have corresponding antonyms in  $D'_i$ .

Clearly, when computing aspect-wise similarity score  $s_{a_i}$ , we do not consider any synonyms or semantic similarity between two description words — the similarity is based solely on the use of exact words. Our strategy of exact word matching is a simplification, but since similarity of adjectives and adverbs is not as well-studied as that of nouns (various similarity metrics for nouns are available for WordNet, but none of those work on adjectives and adverbs), clustering description words is not as trivial as clustering aspects<sup>6</sup>.

#### 5.4 C+N-GRAM+SYN model

To test whether a given review  $r$  about the hotel  $h$  is deceptive or not, we extend Feng et al.’s N-GRAM+SYN model by incorporating our aspect-wise compatibility features into it. We call this extension the C+N-GRAM+SYN model, which includes the following two categories of features:

1. Alignment compatibility features
  - (a) Compatibility features for alignment  $Q \rightarrow P$ , including  $m_{a_i}$ ,  $s_{a_i}$ , and  $c_{a_i}$  for each of the  $N$  aspects  $\{a_1, \dots, a_N\}$  with the highest occurrences in  $P$ .
  - (b) Compatibility features for alignment  $P \rightarrow Q$ , including  $m_{a'_i}$ ,  $s_{a'_i}$ , and  $c_{a'_i}$  for each of the  $N$  aspects  $\{a'_1, \dots, a'_N\}$  with the highest occurrences in  $Q$ .
2. Baseline features: 3,009 unigrams, 10,538 bigrams, and 6,571 syntactic production rules, as described in Section 5.1.

The resulting dimension of our full feature set is  $20,118 + 2N$ , where  $N$  ranges from 10 to 100 in our experiments (see Section 6.2). For all experiments, we use SVM<sup>perf</sup> (Joachims, 2006) to train all the linear SVM classifiers.

## 6 Results and discussion

### 6.1 Reimplementing baseline models

We first demonstrate the performance of our reimplemented baseline models. We conduct 5-fold

<sup>5</sup><http://wordnet.princeton.edu/wordnet/>

<sup>6</sup>In fact, we have experimented with using LSA-based word vectors to cluster description words, but the performance is slightly lower than using exact word matching.

Features	Prec	Rec	$F_1$	Acc
SYN	87.2	88.8	88.0	87.9
N-GRAM	89.7	89.5	89.6	89.6
1-GRAM+SYN	88.0	90.0	89.0	88.9
2-GRAM+SYN	87.9	90.5	89.2	89.0
N-GRAM+SYN	89.2	91.3	90.2	90.1

Table 3: Results (in %) of our reimplemented baseline models. Performance is reported using 5-fold cross-validation over the op\_spam\_v1.3 dataset (800 reviews of 20 hotels).

cross-validation experiments over the 800 reviews of 20 hotels, in which each fold contains all reviews from four hotels, such that the learned models are always tested against unseen hotels. Performance is reported by accuracy and the micro-averaged precision, recall, and F-score of retrieving *deceptive* reviews (see Ott et al. (2013) for details). We experiment with five different baseline feature sets: (1) SYN: syntax features, (2) N-GRAM: unigram and bigram features, (3) 1-GRAM+SYN: unigram and syntax features, (4) 2-GRAM+SYN: bigram and syntax features, and (5) N-GRAM+SYN: unigram, bigram and syntax features, where the second is Ott et al.’s model<sup>7</sup>, and the third is Feng et al. (2012a)’s model.

The results are shown in Table 3. Our reimplemented Ott et al.’s model (N-GRAM) obtains 89.6% accuracy, while the accuracy of our reimplemented Feng et al. model (1-GRAM+SYN) is noticeably lower (88.9%). Though its improvement over the N-GRAM model is not significant, the N-GRAM+SYN model performs the best (90.1%) among all baseline models, and thus is chosen to be the baseline in our later experiments.

### 6.2 Comparison with baseline models

We now demonstrate the performance achieved by our profile alignment compatibility features. We experiment with using our profile alignment features in combination with baseline features, i.e., the C+N-GRAM+SYN model. We first study the

<sup>7</sup>Actually, the best performance reported by Ott et al. was obtained using their LIWC + N-GRAM model, which is the combination of N-GRAM features and 80 features output by the Linguistic Inquiry and Word Count (LIWC) software (Pennebaker et al., 2007). However, since the improvement over  $n$ -gram features alone was small and insignificant (0.2% absolute improvement in accuracy), and we were not able to successfully reproduce the improvement, the comparison with our model is based on using N-GRAM features alone.

effect of the number of reviews used in constructing collective profiles (see Section 5.2.3). We experiment with using the top 50, 100, 150, and 200 reviews written by authors with the highest helpfulness votes (for some hotels, the total number of available reviews is less than 200). For each set of constructed profiles, we tried different values of  $N$ , the number of aspects included in each profile (both collective and test profiles), ranging from 10 to 100, and determine the optimal number by conducting 4-fold cross-validation experiments using 80% of the training data.

The results are shown in Table 4. First, we see that the best overall performance, in terms of  $F_1$  score and accuracy, is achieved using the top 50 or 100 reviews for profile construction. This suggests that using a fairly small amount of reliable data is sufficient to recognize compatibility between collective profiles and test reviews. In fact, using a larger number of reviews is slightly detrimental to the overall performance, since more noise might be included in the process.

Moreover, we see that our best model, achieved using 50 or 100 reviews for profile construction, outperforms the best baseline model, N-GRAM+SYN, on all four metrics, by over 1%, which is a 12.1% error reduction rate. The improvement is confirmed to be statistically significant ( $p < .05$ ) using the Wilcoxon sign-rank test. In addition, we inspect each model’s predictions on individual test instances, and discover that our C+N-GRAM+SYN model correctly classifies some instances on which the baseline model makes an error, and at the same time does not make any additional errors that the baseline model does not make.

Finally, with respect to the value of  $N$ , which is the number of aspects included in each profile, we discover that smaller values of  $N$ , i.e., 10 or 20, consistently give the most satisfying results.

## 7 Conclusions and perspectives

In this paper we proposed the use of profile alignment compatibility as an indicator of truthfulness in product reviews. We defined two types of compatibility between product profiles, and designed a methodology to tackle them by extracting aspects and associated descriptions from reviews. We adopted Ott et al.’s op\_spam.v1.3 dataset of hotel reviews, and improved the N-GRAM+SYN model of Feng et al. (2012a). Our approach was

Reviews used	$N$	Prec	Rec	$F_1$	Acc
50	10	90.2*	92.5*	<b>91.4*</b>	<b>91.3*</b>
100	10	90.0*	<b>92.8*</b>	<b>91.4*</b>	<b>91.3*</b>
150	20	<b>90.6*</b>	92.0*	91.3*	<b>91.3*</b>
200	20	90.4*	91.8*	91.1*	91.0*

Table 4: Results (in %) of our profile alignment compatibility models. Performance is reported using 5-fold cross-validation over the op\_spam.v1.3 dataset (800 reviews of 20 hotels). The best result of each metric is shown in bold face. All numbers are significantly better (denoted by \*) than the baseline in Table 3, using the Wilcoxon sign-rank test ( $p < .05$ ).

shown to significantly improve the performance of identifying deceptive reviews.

In future work, it is critical to know how well our methodology of extracting aspects and their descriptions from hotel reviews generalizes on other kinds of reviews. We suspect that the approach should work equally well on other domains, as long as the aspects are realized mainly by noun phrases, especially that distinct aspects are realized by proper noun phrases.

Another particularly interesting direction is to explore how our C+N-GRAM+SYN model performs on identifying fake *negative* reviews, as recently released by Ott et al. (2013), rather than the *positive* reviews used in this work. While negative opinion spam is more hazardous to a brand’s fame compared to positive ones, and thus identifying fake negative reviews might be more crucial, one potential difficulty of our approach is that genuine extremely negative reviews written by renowned reviewers are much more sparse than extremely positive ones, especially for famous products, such as the most popular Chicago hotels in the op\_spam.v1.3 dataset.

## Acknowledgments

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada and by the University of Toronto. We would like to thank Myle Ott from Cornell University for kindly providing us with the op\_spam.v1.3 dataset.

## References

- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.
- Song Feng, Ritwik Banerjee, and Yejin Choi. 2012a. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175, Jeju Island, Korea, July. Association for Computational Linguistics.
- Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. 2012b. Distributional footprints of deceptive product reviews. In *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM 2012)*, June.
- Stephanie Gokhman, Jeff Hancock, Poornima Prabhu, Myle Ott, and Claire Cardie. 2012. In search of a gold standard in studies of deception. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 23–30, Avignon, France, April. Association for Computational Linguistics.
- Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of First ACM International Conference on Web Search and Data Mining (WSDM-2008)*, pages 219–230.
- Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 1549–1552.
- Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06*, pages 217–226, New York, NY, USA. ACM.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 423–430, Sapporo, Japan, July.
- Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 939–948.
- Decang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, Madison, Wisconsin, USA, July.
- Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*, volume 5 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, May.
- Bin Lu, Myle Ott, Claire Cardie, and Benjamin K. Tsou. 2011. Multi-aspect sentiment analysis with topic models. In *ICDM Workshops*, pages 81–88.
- Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 309–319, Portland, Oregon, USA, June.
- Myle Ott, Claire Cardie, and Jeffrey T. Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Short Papers*, Atlanta, Georgia, USA, June. Association for Computational Linguistics.
- James W. Pennebaker, Cindy K. Chung, Molly Ireland, Amy Gonzales, and Roger J. Booth. 2007. The development and psychometric properties of LIWC2007. Technical report, Austin, TX, LIWC.Net.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.
- Victoria L. Rubin and Tatiana Vashchilko. 2012. Identification of truth and deception in text: Application of vector space model to rhetorical structure theory. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 97–106, Avignon, France, April.
- Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 308–316, Columbus, Ohio, June. Association for Computational Linguistics.
- Guangyu Wu, Derek Greene, Barry Smyth, and Pádraig Cunningham. 2010. Distortion as a validation criterion in the identification of suspicious reviews. In *Proceedings of the First Workshop on Social Media Analytics (SOMA 2010)*, pages 10–13, New York, NY, USA.
- Kyung-Hyan Yoo and Ulrike Gretzel. 2009. Comparison of deceptive and truthful travel reviews. In *Information and Communication Technologies in Tourism 2009*, pages 37–47. Springer Vienna.

# *Behind the Times: Detecting Epoch Changes using Large Corpora*

Octavian Popescu and Carlo Strapparava

FBK-irst, Trento, Italy

{popescu, strappa}@fbk.eu

## Abstract

Using large corpora of chronologically ordered language, it is possible to explore diachronic phenomena, identifying previously unknown correlations between language usage and time periods, or epochs. We focused on a statistical approach to epoch delimitation and introduced the task of epoch characterization. We investigated the significant changes in the distribution of terms in the Google N-gram corpus and their relationships with emotion words. The results show that the method is reliable and the task is feasible.

## 1 Introduction

Traditionally, scholars of history define epochs according to their deep knowledge and understanding of facts over a long stretch of time. Intuitively, in order to define a new epoch, both a big social impact of a series of events and new issues, which arouse the social interest, must be observed. However, it is hard to define what makes a feature “distinctive” or an event a “great change”. It is even harder to evaluate and measure the impact of a series of changes in society in an objective way. Since the advent of regular newspapers and the industry of mass media, written information has represented a mirror of the interests of society. A social event is relevant only if people pay attention to it and comment on it. A major change in society is reflected in the frequencies with which a set of topics is mentioned in mass media, some of them becoming mentioned more often than previously, while some others are no more of interest. Furthermore, specific epochs typically develop a particular form of wording or rhetorical style.

In this paper we describe a computational approach to *epoch delimitation* on the basis of word distribution over certain periods of time. A big

quantity of data, chronologically ordered, allows accurate statistical statements regarding the covariance between the frequencies of two or more terms over a certain period of time. By discovering significant statistical changes in word usage behavior, it is possible to define epoch boundaries. We show that it is possible to distinguish a series of limited periods of time, spanning at most three years, within which non-random changes affect the joint distribution of terms. Between two such short periods (i.e. the boundaries) no statistical significant changes are observed for decades, and thus we can refer to it as an epoch. The distributions of the considered terms before and after boundaries are distinctly different.

We also introduce the task of *epoch characterization*. Certain words carry with them an emotional charge, like *joy*, *fear*, *disgust* etc. Within a given epoch, we can analyze the distribution of emotion words and their co-occurrences with the set of terms considered indicative for epoch definition. The pattern of these co-occurrences constitutes a blueprint of emotional tendencies with respect to some particular topics in the society within a certain period. Given an arbitrary sample of data from a given, but unknown period of time, the task consists in correlating the emotional pattern of the data with the one of an epoch from which the data comes. The experiments reported here show that this task is feasible and sensible results are obtained.

The corpus used in the current experiments is the Google 5-grams made of all tuples of consecutive 5 words, coming from English books printed roughly from 1614 to 2009.

For the purpose of the present paper, we compiled a lexicon of political and social terms. The lexicon contains 761 words, such as: *capitalism*, *civil disobedience*, *demagogue*, *democracy*, *dictator*, *chickenhawk*, *education*, *government*, *peace*, *war* etc. These terms come from the lists



compiled for the political and sociological domain publicly available<sup>1</sup>. The frequency of these terms and their covariance is analyzed over the years and non-random changes are found according to the methodology presented in Section 3. The methodology itself is purely statistical and it does not depend in any way on what the list contains. We could have equally chosen terms from art or sport domain, obtaining epoch boundaries specific to each domain.

The emotion words used in epoch characterization come primarily from the NRC Word-Emotion Association Lexicon (Mohammad and Turney, 2010) to which the list of emotion words extracted from WordNet-Affect (Strapparava and Valitutti, 2004), distributed in the Semeval 2007 Affective Text task (Strapparava and Mihalcea, 2007), has been added. The lexicon is made up of English words to which eight possible tags are attached: *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise* and *trust*. All in all there are 14,000 words for which at least one affective tag is given.

The paper is organized as follows. In Section 2 we review the relevant literature. Section 3 presents the statistical apparatus employed in epoch determination and epoch characterization. In Section 4 we present the experiments and the results we have obtained. In the last section we highlight the contribution of this paper and make an overview of further immediate work.

## 2 Related Work

In (Michel et al., 2011), besides a complete introduction to the Google Books corpus, a limited diachronic study of words meaning and form is also carried out. The authors introduce the term ‘culturomic’ and show that quantitative analyses may lead to interesting results. They show that it is possible to determine censorship and suppression by comparing the frequencies of proper names in bilingual Google books corpora. However, the authors did not proceed to a systematic studies of epochs.

Regarding semantic change, the task of sense disambiguation over the years is introduced in (Mihalcea and Nastase, 2012). In their paper, the authors refer to definite periods of time as epochs but they considered them prior defined.

In (Wang and McCallum, 2006) an analysis of topics over time is carried out. The paper fo-

cuses on rather fixed topics, which are expressed by frozen compounds, such as “mexican war”, “CVS operation”, and determines how these topics evolve during the years. However, because the scope of their paper is not global, the corpus used comes from 19 months of personal emails. It is hard to see how this method could generalize. A similar approach is described in (Wang et al., 2008). The authors use LDA to facilitate the search into large corpora by automatically organizing them.

In (Yu et al., 2010), the statistics tests and the google N-gram corpus are used for (semi) automatic creation and validation of a sense pool. The frequencies extracted from Google N-gram corpus are filtered with an appropriate statistical test and further verified by human experts.

The richness and complexity of cultural information contained in the Google N-gram corpus is analyzed in (Joula, 2012). By considering the degree of interdependence as a measure for complexity, the author used the 2-gram corpus to analyze the complexity of American culture. However, there is no the epoch distinction and statistical support.

Regarding Sentiment analysis, text categorization according to affective relevance, opinion exploration for market analysis, etc. are just some examples of application this NLP area (Pang and Lee, 2008). While positive/negative valence checking is an active field of sentiment analysis, a fine-grained emotion checking is nowadays an emerging research topic. For example, SemEval task on Affective Text (Strapparava and Mihalcea, 2007) focussed on the recognition of six emotions in a corpus of news headlines.

## 3 Methodology

In this section we present the statistical tests we used to analyze the data. We do not assume any prior distribution of the frequencies in the corpus and we employ both non parametric and parametric tests. In this section we present the statistical tests we used to analyze the data. We do not assume any prior distribution of the frequencies in the corpus and we employ both non parametric and parametric tests. The Google N-gram corpus is made up of a number of text files which contain N-grams, where N goes from 1 to 5, and which are obtained from English books published over the years. In Table 1 we present a snippet from the

<sup>1</sup>E.g. [www.democracy.org.au/glossary.html](http://www.democracy.org.au/glossary.html)

5-gram corpus:

<i>n-grams</i>	<i>year</i>	<i># occ.</i>	<i># pages</i>	<i># books</i>
democracy at work	1996	1	1	1
democracy at work	1997	5	5	5
democracy at work	1998	2	2	2

Table 1: 5-Gram Google files

### 3.1 Statistical tests

**Normalization.** Due to the exponential growth of the published data, it is better to normalize the number of occurrences for a meaningful comparison. We considered all the content nouns, including proper names, and we computed for each term of interest the percentage of occurrences of that term with respect to the sum of frequencies of all content nouns (considering lemmata). In this paper, when we refer to frequency of a term we mean the normalized figure, unless explicitly stated otherwise. The percentage is in fact very informative on what the public opinion is concerned about in certain periods and substantial differences may be observed within a short period of time. For example, *democracy* was 25 times less a probable topic at the begin of twenty-first century than 50 years before. In such cases, one can clearly talk about a change of interest in society, see Figure 1.

**Welch’s test.** Welch test is a variant of t-student test to check whether two different samples come from the same population or not (Sawilowsky, 2001). The Welch test fits our purposes because it does not assume that the sample have equal variance, thus it can be applied where the other similar tests, such as classical t-student or F-test, do not. The initial conditions for Welch test does not include (1) the equality of the sample sizes and (2) either the homogeneity of population, thus the data may not come from a population having a distribution with a unique variance. In fact for this reason we prefer to use non parametric test in the present paper.

In practice, we apply the Welch’s test to sample size representing contiguous periods of time . To exemplify, let us consider here the term “war” and two different periods 1800-1900 and 1900-2000. Each period is split in two sub-periods, 1800-1850 vs. 1850-1900, and 1900-1950 vs. 1950-2000 respectively. We test whether the samples 1800-1850 vs. 1850-1900 have the same mean, and we also test whether the samples 1900-1950 vs. 1950-

2000 have the same mean. In Table 2 we present the results obtained.

<i>Sample</i>	<i>t</i>	<i>Outcome</i>
1800-1850 vs. 1850-1900 $\mu_1 = .078$ vs. $\mu_2 = .081$	0.23	No Rejection at $\alpha = 0.1$
1900-1950 vs. 1950-2000 $\mu_1 = .184$ vs. $\mu_2 = .098$	-5.163	Rejection at $\alpha = 0.01$

Table 2: Welch’s test for term *war*.

The null hypothesis, that the two sample come from a population with the same mean cannot be rejected at  $\alpha = 0.1$  in the first case. The same null hypothesis is rejected with a very high confidence,  $\alpha = 0.01$  in the second case.

**Run Test.** Run test is a non parametric test, which determines whether the a sequence of numbers is likely to be the result of a random process or there might be an inner pattern in data (Gibbons and Chakraborti, 1992; Lindgren, 1993). For example let us suppose that we have a Bernoulli process with “+” and “-” possible outcomes and probabilities 1/4, 3/4 respectively. A sequences like ++++-----+----- is very unlikely to be a random generated sequences of this process. The run test is designed to detect such cases. A set of real values, as the frequencies of a term over a period of time are, is converted into a run sequences by considering the median of the sequence and obtaining a new sequences by marking with a “+” if the value is bigger than the median and with a “-” if not.

In practice we apply the run statistics on frequencies of a set of given terms. For example for the term *government*, considering two periods we obtain the results in Table 3, where  $p_1=1800-1850$ ,  $p_2=1850-1900$ , and  $p_3=1900-1950$ .

	<i>run</i>	<i>run-test</i>
$p_1$	+++-----++-++++-----	.29839
$p_2$	+++++++-----+-----	.32603
$p_3$	-----+-----+-----	.00001

Table 3: Run test for term *government*

The null hypothesis, which is that the run sequence is randomly generated can be rejected at a significance level  $\alpha = 0.1$  for the third sample, namely from 1900-1950.

**Least Squares.** The least squares method is used to find the line with the smallest sum of square of the difference between the data and the line points ,(Björck, 1996).

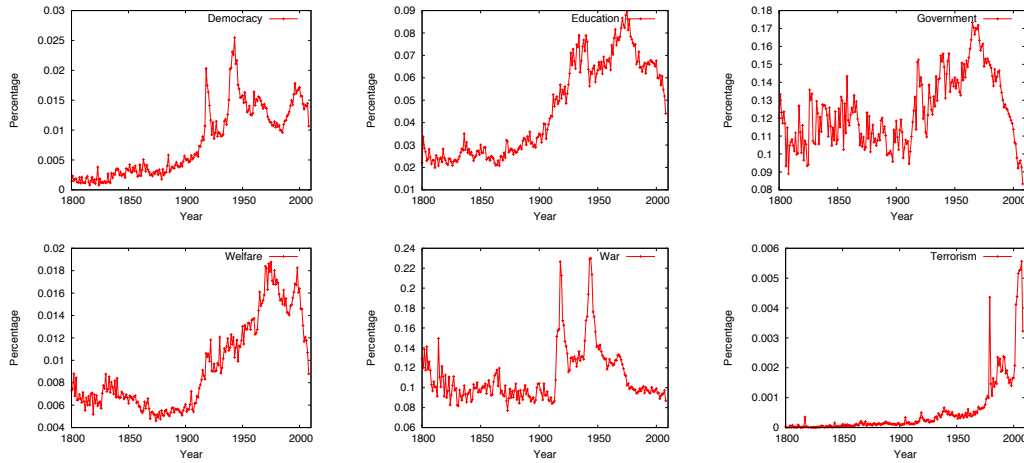


Figure 1: *democracy, government, education, welfare, war and terrorism* percentage

In practice we try to determine the longest period of time in which the data could be fit to a line, imposing that the sum of squares is bond by a small value. For example least squares method applied to the term *government* from 1968 to 2008 produce the optimal line plotted in Figure 2. The line has the equation:  $y = 3.807 - 0.001x$ . The sum of residuals is less than 0.002 ( $ss = 0.0014$ ), which means that the average variance around the line points is 0.00036. This represents a remarkable fit of data to a line.

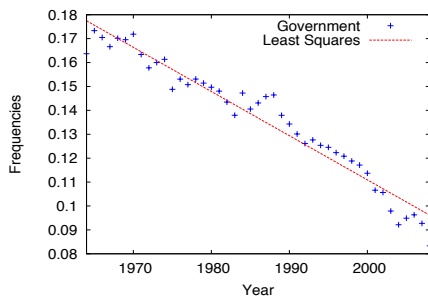


Figure 2: Least squares applied to the frequencies for term *government*

**Ratio.** It is usual to find in the distribution of frequencies increasing or decreasing sequences. For a definite period of time where a particular direction of growth is observed, we take into account also the rate of growth defined as the ratio between the difference of a three consecutive values:  $\frac{(x_i - x_{i-1})}{(x_{i+1} - x_i)}$ . In practice we use the growth ratio for (C1) characterizing a whole period of time and, (C2) for detecting similarities among distributions for different/same terms over the same/different periods.

**C1** The same growth rate may characterize a whole period of time. A change in the growth rate may signal the beginning of a new epoch. In Table 4, we report the median growth rate for the term *democracy* over two periods.

year	growth rate series						average
1850-1900	1.119	1.227	1.227	1.231	1.136	1.23	1.183
1900-1940	1.218	1.298	1.559	1.69	1.751	1.791	1.802

Table 4: Growth Rate for *democracy* over two periods

**C2** Considering the difference of frequencies of two terms and using the run test we can observe if the growth rate remain the same or changed. In Table 5 we present two runs from two different period of times for the ratio of the differences between the terms *education* and *democracy*. We observe that we have the same growth ratio pattern in different periods.

year	growth rate of difference	
1850-1900	+	+++++
1900-1950	+	+++++

Table 5: Growth rates patterns of the difference between *education* and *democracy*

**Spearman and Kendall Test.** Spearman and Kendall tests are two non parametrical tests for measuring the statistical dependencies between two variables. In practice the time line is always one of the variable and a positive answer from one of this tests shows a non-random evolution of the frequencies within a that period of time. Usually we consider the difference between the frequencies of two terms and apply the Spearman and Kendall test against the timeline. In Table 6 we

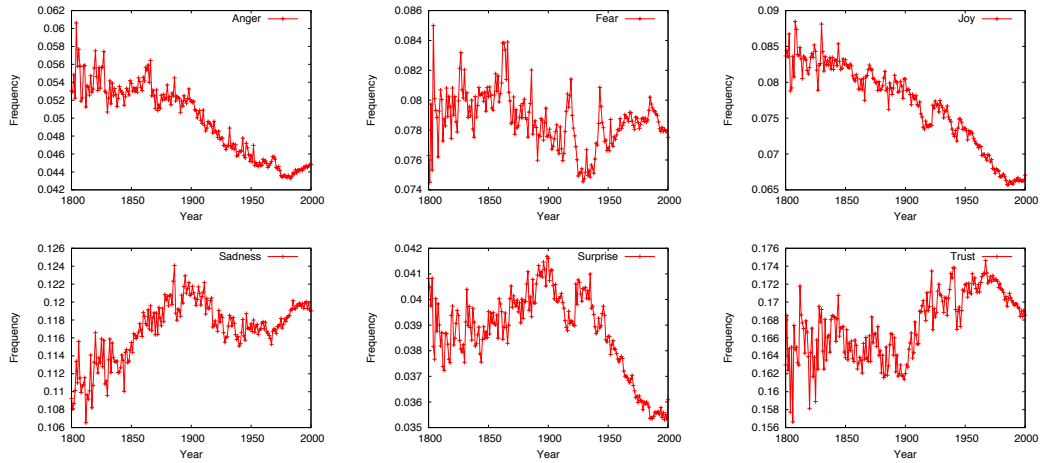


Figure 3: *anger, fear, joy, sadness, surprise and trust* percentage

give an example of the output of the two tests applied to difference between *government* and *welfare*.

year	Spearman	Kendall
1800-1850	0.9689	0.8530
1850-1900	0.7243	0.5510
1900-1950	-0.293	-0.2081
1950-2000	0.7493	0.5934

Table 6: Spearman and Kendall test for time vs. difference between *government* and *welfare*

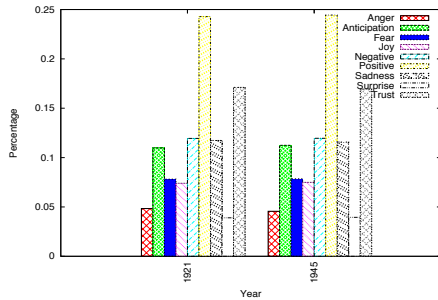


Figure 4: Emotion percentages in 1921 and 1945

The results above show that from the point of view of the relationship between the frequencies of *government* and *welfare* we can clearly distinguish four different patterns. There is a strong statistical evidence that the frequencies two terms were correlated in the period 1800-1850 and independent between 1900-1950.

Before concluding this section we also plot the frequencies of the emotion terms and two examples of emotion blue-print for years 1921 and 1945. The counts were normalized taking into account the emotion words (see Figures 4 and 3).

### 3.2 Epoch: Decision Procedure

In Section 3.1 we presented the statistical procedures we use for epoch determination. Each of these tests is able individually to find non-random changes in the distribution of the frequencies of terms over the years and to find the beginning and the end of the time periods where the same statistically relevant pattern - linear, same growth rate, dependency - is observed. However, noticing a change in the distribution is not enough for declaring the begin or the end of an epoch. The fact that many of the terms considered are affected by a change in their distribution more or less concomitantly must be observed in order to decide on the epoch boundaries. For now, we preferred a conservative view therefore in the experiments we carried we impose that significantly more than 50% of the terms change their distribution and that the period in which this is happening is at most three years. The algorithm for epoch determination using the tests introduced above is:

---

#### Algorithm Epoch Detection

---

**Require:** Google N-grams with time info

**Ensure:** Epoch

- 1: Apply *Welch's* and *Run* test for non-random changes
  - 2: Choose *start\_year* and *end\_year* spanning several decades
  - 3: **if** number of terms positive to line 1 tests in the time interval  $\pm 3$  years around *start\_year* and *end\_year*  $\leq 50\%$  **then**
  - 4:     **goto** line 2
  - 5: **end if**
  - 6: Apply Least Square, Ratio, Spearman and Kendall
  - 7: **if** number of terms positive to line 6 tests  $\leq 50\%$  **then**
  - 8:     **goto** line2
  - 9: **end if**
  - 10: epoch  $\leftarrow [start\_year, end\_year]$
-

At step 6, the order in which the tests are applied is exactly as specified. If Least Square is positive then also the others are positive as well. An so on: if Ratio holds also the last two tests hold. Condition 7 is satisfied if at least Kendall is positive.

## 4 Experiments

We considered a list of 761 political terms and we applied the decision procedure presented in Section 3.2. The output of the decision procedure is a set of years around which statistically significant changes in the distribution of frequencies for the majority of the terms considered occur. The epoch identified for the chosen list of terms and the decision procedure detailed in Section 3 identified the following 6 epochs epochs between 1800 and 2009, see Table 7.

<i>epoch 1</i>	1800-1860	<i>epoch 4</i>	1950-1975
<i>epoch 2</i>	1860-1900	<i>epoch 5</i>	1975-1999
<i>epoch 3</i>	1900-1950	<i>epoch 6</i>	1999-2009

Table 7: Epochs between 1800-2009

<i>term</i>	<i>change year</i>	<i>positive test</i>
two party system	1975	run, ratio
two party system	1999	Welch's, ratio
patriotism	1975	Welch's, ratio
patriotism	1999	Welch's, squares
too big to fail	1975	ratio
too big to fail	1999	squares

Table 8: Statistical significant changes

Table 8 lists a few examples of terms affected by a statistical change at epoch boundaries. In Table 9 we present the number of terms which changed their distribution for each boundary, on the second column the absolute value and on the third column the percentage relative to the total number of terms considered, 761. We can see that the number of terms which are positive to statistical tests varies substantially. However, it is not by chance that the changes occur.

<i>year</i>	<i>number of terms</i>	<i>percentage</i>
1860	518	68%
1900	491	64%
1950	579	76%
1975	682	89%
1999	607	78%

Table 9: The number of terms defining an epoch

There is a tolerance of a couple of years around the boundaries. For example, if a term's distribution changes  $\pm 3$  years around 1975, then this

change is considered for epoch boundary delimitation. Especially in the last 60 years, it seems that the changes occur more frequently and they are more clearly delimited. During these times, the changes between two different trends occur within a couple of year in the great majority of cases. The dynamic of change is different in the nineteenth century, when it is more likely to observe a buffer zone for several years. In the buffer zone, the distribution around a mean value is quasi normal.

In fact, by running Spearman and Kendall tests we discovered interesting dependencies between the distribution of certain terms and the time line. We computed the differences between the frequencies of pair of terms. For example, for the pair *socialism* and *capitalism* the results of the statistical tests show a strong correlation within each epoch, see Table 10 and Figure 5.

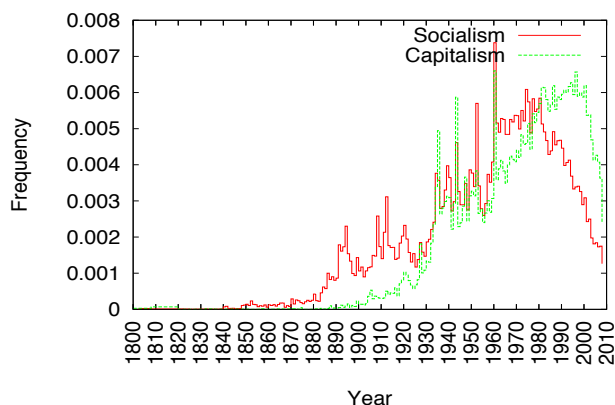


Figure 5: Socialism vs. Capitalism through the epochs

<i>epoch</i>	<i>Spearman Test</i>	<i>Kendall Test</i>	<i>Dependent</i>
1800-1860	0.9741	0.9138	yes
1860-1900	0.9402	0.8429	yes
1900-1950	0.2073	0.0108	no
1950-1975	0.2210	0.0962	no
1975-1999	-0.9762	-0.8977	yes
1999-2009	-0.945	-0.8891	yes

Table 10: *socialism* vs. *capitalism* through the epochs

<i>anger</i>	<i>anticipation</i>	<i>disgust</i>	<i>fear</i>
3914	9390	2448	6519
<i>joy</i>	<i>sadness</i>	<i>surprise</i>	<i>trust</i>
6053	9892	3173	12082

Table 11: Emotion words in Google 5-grams ( $\times 10^6$ )

term	1800-1860	1860-1900	1900-1950	1950-1975	1975-1999	1999-2009
anger	0.0546	0.0540	0.0491	0.0467	0.0458	0.0455
anticipation	0.1093	0.1112	0.1129	0.1157	0.1178	0.1158
disgust	0.0358	0.0344	0.0303	0.0282	0.0283	0.0287
fear	0.0813	0.0813	0.0786	0.0813	0.0819	0.0769
joy	0.0842	0.0818	0.0771	0.0736	0.0693	0.0706
sadness	0.1149	0.1224	0.1206	0.1216	0.1240	0.1188
surprise	0.0395	0.0421	0.0405	0.0388	0.371	0.0378
trust	0.1680	0.1680	0.1722	0.1791	0.1775	0.1672

Table 12: The average of emotion frequencies over the epochs

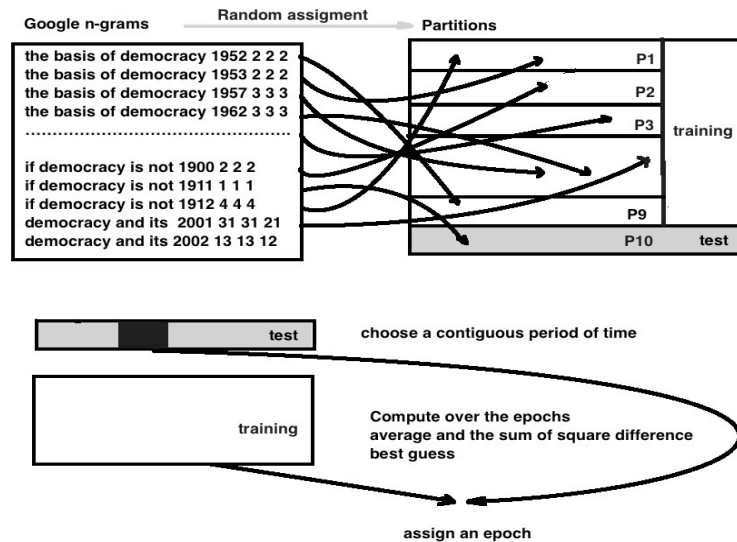


Figure 6: 10-fold validation

To each epoch an emotional blueprint can be attached. An emotional blueprint is obtained by taking into consideration the emotion denoting terms. There are 7 emotion words; *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise*, *trust* and two opinion words, *negative* and *positive*. The corpus we consider in this section is the part of Google 5-grams in which each 5-gram contains at least an emotion word. In Table 11 we present their distribution in Google-gram corpus.

The epoch characterization task consists in using the epochs as categories and assigning an unseen sample covering a continuous, but unknown, period of time to one of the categories. For the experiments in this paper, we used the average values of each emotion term computed over the epochs as epoch blueprint, thus each epoch is characterized by a unique value for each emotion term, see Table 12.

For evaluation we used a k-fold cross validation approach. The k-partitions were obtained by

choosing randomly for each occurrence in google corpus its partition, so in average each partition had an equal number of terms. The training was carried on  $k - 1$  partitions and tested on a single partition, thus there are  $k$  independent evaluation experiments. The training  $k - 1$  partitions were joined into a unique corpus which was split into epochs and for each epoch we computed the average for each emotion term. The test partition, the k-partition was split in ten contiguous sub-partitions. For each test sub-partition, the average of the emotion terms was computed and compared against the averages from training corpus to find the most similar ones, resulting in  $10k$  experiments (see Figure 6).

The procedure of finding the most similar epoch can be implemented in different ways. We discuss here two approaches. The first method computes the average over the training corpus for each emotion term and, separately, the average for the test corpus and sums up the squares of the differences

experiment	first run	second run	third run	fourth run	fifth run
all occurrences squares sum	46%	51%	46%	48%	50%
all occurrences best guess	60%	56%	60%	59%	60%
co-occurrences squares sum	53%	58%	59%	57%	59%
co-occurrences best guess	65%	69%	67%	66%	66%

Table 13: 5-partition cross-validation results

for each particular epoch. The category assigned is the one with the least sum of squares. The second method compares the averages computed over the training for each epoch and chooses a representative for each epoch, let us call it best guess. The test sample compares only the averages against the best guess for each epoch and it is assigned to the epoch which has the closest best guess.

To measure the accuracy, we simply count how many times there was only one epoch chosen and that it was indeed the correct one. The figures reported in Table 13 represent the accuracy, as all the sub-partitions were checked and consequently the recall was 1. The last two experiments we carried out on considered political terms. Instead of considering all occurrences of the emotion terms inside a particular epoch, we considered only the co-occurrences of the emotion words with a set of political terms. For this purpose we chose a set of 20 from the list of 761 of political terms considered: *capitalism, community, common good, democracy, education, free market, government, heresy hunting, individual rights, justice, middle class, money, nepotism, politics, public interest, savings, socialism, social system, technology, and war*. The averages for each corpus, training and test respectively, were computed only for these terms and the two approaches above, squares sum and best guess were applied.

In order to understand whether the results above are informative, we run a simple baseline over the same data. The baseline decision was to consider for each subpartition a random epoch. The accuracy of the baseline is around 15%.

## 5 Conclusion and Further Research

The possibility to analyze automatically the changes over the time in the usage of certain terms is an open window into sociological studies carried from a language perspective with computational methods.

During the experiments, some interesting research directions have been revealed. Firstly, although we made no attempt here to make the con-

nection between certain changes and real historical events, it seemed that this was indeed possible. Sharply distinctive changes are observed for certain terms around global war dates. Secondly, while we used the ratio as a parameter which may signal a change, we carried no analyses on the typology of rates themselves. Such analyses may bring to light patterns into the dynamic of interests within a society. Thirdly, the methodology we presented can be easily used for prediction. Such studies could predict future changes. A striking example is represented by the covariance between socialism and capitalism, which seemed to indicate the collapse of political regimes in East Europe several years before it actually happened, see Figure 5. We plan to investigate further the distribution of terms over the time going in the directions above.

## References

- A. Björck. 1996. *Numerical Methods for Least Squares Problems*. SIAM: Society for Industrial and Applied Mathematics.
- J. D. Gibbons and S. Chakraborti. 1992. *Nonparametric Statistical Inference*. CRC Press.
- P. Joula. 2012. Using the google ngram corpus to measure cultural complexity. In *Proceedings of Digital Humanities*, University of Hamburg, July.
- B. Lindgren. 1993. *Statistical Theory*. Chapman and Hall/CRC.
- J. B. Michel, Y.K. Shen, A.P. Aiden, A. Veres, M. K. Gray, J.P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, M. A. Nowak S. Pinker, and E.L. Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, January.
- Rada Mihalcea and Vivi Nastase. 2012. Word epoch disambiguation: Finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 259–263, Jeju Island, Korea, July. Association for Computational Linguistics.

- Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA, June. Association for Computational Linguistics.
- B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.
- S. Sawilowsky. 2001. Fermat, schubert, einstein, and behrens-fisher: The probable difference between two means when  $\sigma_1^2 \neq \sigma_2^2$ . *Journal of Modern Applied Statistical Methods*, 1(2):461–472.
- C. Strapparava and R. Mihalcea. 2007. SemEval-2007 task 14: Affective Text. In *Proceedings of SemEval-2007*, Prague, Czech Republic, June.
- C. Strapparava and A. Valitutti. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon.
- X. Wang and A. McCallum. 2006. Topics over time: A non markov continuous-time model of topical trends. In *Proceedings of KDD-06*, Philadelphia, Pennsylvania, August.
- C. Wang, D. Blei, and D. Heckerman. 2008. Continuous time dynamic topic models. In *Proceedings of the International Conference on Machine Learning*.
- Liang-Chih Yu, Chung-Hsien Wu, Ru-Yng Chang, Chao-Hong Liu, and Eduard H. Hovy. 2010. Annotation and verification of sense pools in ontonotes. *Information Processing and Management*, 46(4):436–447, July.



# How Noisy Social Media Text, How Different Social Media Sources?

Timothy Baldwin,<sup>♠♥</sup> Paul Cook,<sup>♥</sup> Marco Lui,<sup>♠♥</sup> Andrew MacKinlay<sup>♠♥</sup> and Li Wang<sup>♠♥</sup>

♠ NICTA Victoria Research Laboratory

♥ Department of Computing and Information Systems, The University of Melbourne

tb@ldwin.net, paulcook@unimelb.edu.au, mhlui@unimelb.edu.au,

Andrew.MacKinlay@nicta.com.au, li.wang.d@gmail.com

## Abstract

While various claims have been made about text in social media text being noisy, there has never been a systematic study to investigate just how linguistically noisy or otherwise it is over a range of social media sources. We explore this question empirically over popular social media text types, in the form of YouTube comments, Twitter posts, web user forum posts, blog posts and Wikipedia, which we compare to a reference corpus of edited English text. We first extract out various descriptive statistics from each data type (including the distribution of languages, average sentence length and proportion of out-of-vocabulary words), and then investigate the proportion of grammatical sentences in each, based on a linguistically-motivated parser. We also investigate the relative similarity between different data types.

## 1 Introduction

Various claims have been made about social media text being “noisy” (Java, 2007; Becker et al., 2009; Yin et al., 2012; Preotiuc-Pietro et al., 2012; Eisenstein, 2013, *inter alia*). However, there has been little effort to quantify the extent to which social media text is more noisy than conventional, edited text types. Moreover, social media comes in many flavours — including microblogs, blogs, and user-generated comments — and research has tended to focus on a specific data source, such as Twitter or blogs. A natural question to ask is how different the textual content of the myriad of social media types are from one another. This is an important first step towards building a general-purpose suite of social media text processing tools.

Most research to date on social media text has used very shallow text processing (such as

keyword-based time-series analysis), with natural language processing (NLP) tools such as part-of-speech taggers and parsers tending to be disfavoured because of the perceived intractability of applying them to social media text. However, there has been little analysis quantifying just how hard it is to apply NLP to social media text, or how intractable the data is for NLP tools.

This paper addresses the two issues above. We build corpora from a variety of popular social media sources, including microblogs, user-generated comments, user forums, blogs, and collaboratively-authored content. We then compare these corpora to more conventional texts through a variety of statistical and linguistic analyses to quantitatively assess the relative extent to which they are “noisy”, and quantify similarities between them. Our findings indicate that there are certainly differences between social media sites, but that if we focus our attention on English text, there are striking similarities, and that even sources such as Twitter may be more “NLP-tractable” than they are often portrayed.

## 2 Background

Natural language processing (NLP) has been applied to a wide range of applications on social media, especially Twitter. Numerous studies have attempted to go beyond simple keyword and burstiness models to identify real-world events from Twitter (Benson et al., 2011; Ritter et al., 2012; Petrovic et al., 2012). Recent efforts have considered identifying user location based on the textual content of tweets (Wing and Baldrige, 2011; Roller et al., 2012; Han et al., 2012b) and user metadata (Han et al., 2013). Related work has examined models of the relationships between words and locations for the purpose of identifying and studying regional linguistic variation (Eisenstein et al., 2010; Eisenstein et al., 2012).

Given the abundance of non-standard language

on social media, including lexical variants (e.g. *supa* for *super*) and acronyms (e.g. *smh* for *shaking my head*), as well as genre-specific phenomena such as the usage of hashtags and mentions on Twitter, standard NLP tools cannot be immediately applied. Efforts to address this problem have taken two main approaches: modifying social media data to more closely resemble standard text, and building social media-specific tools.

Lexical normalisation is the task of converting non-standard forms such as *tlkin* and *touchdoown* to their standard forms (*talking* and *touchdown*, respectively), in the hopes of making text more tractable to NLP (Eisenstein, 2013). Approaches to normalisation have exploited various sources of information including the context in which a given instance of a lexical variant occurs (Gouws et al., 2011; Han and Baldwin, 2011), although the best results to date have been achieved by automatically discovering lexical variant–standard form pairs from a large Twitter corpus (Han et al., 2012a). This latter approach is particularly appealing because it allows for very fast normalisation, suitable for processing large volumes of text.

Conversely, Owoputi et al. (2013) and Ritter et al. (2011) developed part-of-speech (POS) taggers for Twitter that are better able to handle properties of this text type such as the higher out-of-vocabulary rate compared to conventional text. Ritter et al. further developed a Twitter shallow parser and named-entity recogniser. Foster et al. (2011) evaluated standard parsers on social media data, and found them to perform particularly poorly on Twitter, but showed that their performance can be improved through a retraining strategy.

Another natural question to ask is how similar the characteristics of social media text are to those of other domains. More specifically, we may be interested in a numerical measurement of how closely the language used in one corpus matches that of another. Kilgarriff (2001) proposed a method for calculating both inter-corpus similarity and intra-corpus homogeneity, and language modelling has also been used as the basis for calculating how well one corpus models another. We discuss both of these options below.

### 3 Datasets

In order to evaluate the characteristics of text in different social media sources, we assembled the

following datasets from across the spectrum of popular social media sites, varying in terms of document length, the number of authors/editors per document, and the level of text editing:

**TWITTER-1/2:** micro-blog posts from Twitter, crawled using the Streaming API over two discrete time periods (TWITTER-1 = 22 September 2011 and TWITTER-2 = 22 February 2012) to investigate the temporal-specificity of the data — documents up to 140 characters in length, single author per document, and no facility for post-editing

**COMMENTS:** comments from YouTube, based on the dataset of O’Callaghan et al. (2012), but expanded to include all comments on videos in the original dataset<sup>1</sup> — documents up to 500 characters in length, single author per document, and no facility for post-editing

**FORUMS:** a random selection of posts from the top-1000 valid vBulletin-based forums in the Big Boards forum ranking<sup>2</sup> — documents of variable length (with a site-configurable restriction on maximum post length), single author per document, and optional facility for post-editing (depending on the site configuration)

**BLOGS:** blog posts from tier one of the ICWSM-2011 Spinn3r dataset (Burton et al., 2011) — generally no restriction on length, single author per document, and facility for post-editing

**WIKIPEDIA:** text from the body of documents in a dump of English Wikipedia — no restriction on document length, usually multiple authors/editors per document, and facility for post-editing

As a reference corpus of English from a non-social media source, we also include documents from the British National Corpus (Burnard, 2000):

**BNC:** all documents from the written portion of the British National Corpus (BNC) — documents of up to 45K words from a variety of sources, mostly by a single author, with editing.

We present the number of documents and average document size for each dataset in Table 1.

<sup>1</sup>We post-processed the retrieved comments to remove all occurrences of the unicode U+FEFF codepoint (which is used either as a byte order marker at the start of messages or a zero-width no-break space when used elsewhere in a document), as it skewed the results of the language identification.

<sup>2</sup><http://rankings.big-boards.com>

Corpus	Documents	Average words per document
TWITTER-1	1 000 000	11.8 ± 8.3
TWITTER-2	1 000 000	11.6 ± 8.1
COMMENTS	874 772	15.8 ± 18.6
FORUMS	1 000 000	23.2 ± 29.3
BLOGS	1 000 000	147.7 ± 339.3
WIKIPEDIA	200 000	281.2 ± 363.8
BNC	3141	31 609.0 ± 30 424.3

Table 1: Number of documents and average document size (mean±standard deviation, in words) for each dataset

TWITTER-1/2 and COMMENTS, predictably, contain the shortest documents, with 12–16 words per document on average. Forum posts are around twice the length on average (but the spread of document lengths is considerably greater). Blog posts, on average, contain around ten times the number of words of a forum post, with a greater spread again of document lengths and longer sentences. Amongst our social media sources, Wikipedia documents are by far the longest, but considerably shorter than BNC documents.

## 4 Corpus Pre-processing

We first pre-process each dataset using the following standardised methodology.<sup>3</sup> In the case that the corpus comes with tokenisation and POS information, we strip this and perform automatic pre-processing to ensure consistency in the quality and composition of the tokens/tags.

We first apply `langid.py` (Lui and Baldwin, 2012) — an off-the-shelf language identifier — to each document to detect its majority language. We then extract all documents identified as English for further processing.

We next perform sentence tokenisation. In line with the findings of Read et al. (2012a) based on experimentation with a selection of sentence tokenisers over user-generated content, we sentence-tokenise with `tokenizer`.<sup>4</sup>

Finally, we tokenise and POS tag the datasets using `TweetNLP 0.3` (Owoputi et al., 2013).

One particularly important property of `TweetNLP` is that it identifies content such as mentions, URLs, and emoticons that aren’t typically syntactic elements of a sentence. More-

<sup>3</sup>Acknowledging that superior domain-specific approaches exist, e.g. for Wikipedia sentence tokenisation using markup (Flickinger et al., 2010).

<sup>4</sup><http://www.cis.uni-muenchen.de/~wastl/misc/>

over, it is able to distinguish between usages of hashtags which are elements of a sentence, and those which are not, as in the case of Examples (1) and (2) below, respectively.

(1) love this #awesome view out of my window

(2) Swinging with the besties! #awesome

We POS tag each sentence in each corpus using `TweetNLP`, and remove all tokens identified as non-linguistic.<sup>5</sup> In our examples above, e.g., we remove the token `#awesome` from (2) but not (1).

To normalise for corpus size, we extract a random sample of sentences totalling 5M tokens from each dataset, and further partition this sample into 5 equal-sized sub-corpora.

## 5 Analysis

In this section, we analyse the characteristics of the language used in the respective data sources.

### 5.1 Language Mix

First, we analyse the breakdown of languages found in each data source based on the predictions of `langid.py`, as detailed in Table 2. Note that these results are based on the full datasets without language filtering. Also note that WIKIPEDIA and the BNC are intended to be monolingual English collections, and that FORUMS has a strong bias towards English due to the crawling methodology. For the remainder of the datasets, we expect the results to be representative of the language bias of the respective data sources.

All data sources are dominated by English documents, although in the case of TWITTER-1/2, less than half of the documents are in English (`en`), with Japanese being the second most popular language, and strong representation from languages such as Portuguese (`pt`), Spanish (`es`), Indonesian (`id`), Dutch (`nl`) and Malay (`ms`). These results are largely consistent with earlier studies on the language distribution in Twitter (SemioCast, 2010; Hong et al., 2011).

That the BNC is predicted to be 100% English is a validation of the accuracy of `langid.py`. WIKIPEDIA is more interesting, with tiny numbers (around 0.2% in total) of documents which are predicted to have a majority language of Latin (`la`), German (`de`), etc. Manual analysis of these

<sup>5</sup>Specifically, we remove any token tagged as `#`, `@`, `~`, `U`, or `E`.

TWITTER-1		TWITTER-2		COMMENTS		FORUMS		BLOGS		WIKIPEDIA		BNC	
en	.406	en	.439	en	.757	en	.914	en	.784	en	.998	en	1.000
ja	.144	ja	.124	de	.034	de	.016	ru	.050	la	.000		
pt	.098	es	.091	es	.028	es	.011	fr	.025	de	.000		
es	.093	pt	.072	fr	.023	ro	.009	zh	.022	fr	.000		
id	.031	id	.029	ru	.023	it	.007	de	.019	es	.000		
nl	.025	nl	.022	pt	.020	nl	.007	es	.017	no	.000		
ms	.016	ar	.019	pl	.012	fr	.006	ja	.010	he	.000		
ko	.015	ko	.018	ar	.011	pl	.003	it	.010	zh	.000		
de	.015	ms	.015	it	.011	da	.002	pt	.009	ja	.000		
it	.013	fr	.015	nl	.006	sv	.002	sv	.008	pt	.000		

Table 2: Top-10 languages (by ISO-639-1 identifier) in each dataset

documents reveals that most are made up of lists of different types: names of people from a variety of ethnic backgrounds, foreign place names, or titles of artworks/military honours in various languages. As such, the language tags are actually overwhelmingly correct,<sup>6</sup> in the sense that the predominant language is indeed that indicated.

The implications of these results for text processing of social media are profound. While English clearly dominates the data, there are significant amounts of non-English text in all our social media sources, with Twitter being the most extreme case: the majority of documents are *not* English. Additionally for TWITTER-1/2 and COMMENTS, instances of all 97 languages modelled by `langid.py` were found in the dataset. At the very least, this underlines the importance of language identification as a means of determining the source language in cases where language-specific NLP tools are to be used.

## 5.2 Lexical Analysis

Next, we analyse the lexical composition of the English documents. Hereafter, we focus exclusively on the 5M token subsample of each dataset.

In Table 3 we present simple statistics on the average word length (in characters) and average sentence length (in words) for each dataset. We also analyse the relative occurrence of out-of-vocabulary (OOV) words, based on the GNU `aspell` dictionary v0.60.6.1 with case folding. We strip all “online-specific” markup (hashtags, user mentions and URLs), on the basis of the output of the POS tagger (i.e. any hashtags etc. that are *not* part of the syntactic structure of the text are removed).<sup>7</sup> To filter out common mis-

<sup>6</sup>With the notable exception of Latin, where many of the documents contain lists of names from a variety of European language backgrounds, but little that is identifiable as Latin.

<sup>7</sup>This step reduced the OOV rate in TWITTER-1/2 by

Corpus	Word length	Sentence length	%OOV	
			-norm	+norm
TWITTER-1	3.8±2.4	9.2±6.4	.246	.225
TWITTER-2	3.8±2.4	9.0±6.3	.240	.222
COMMENTS	3.9±3.2	10.5±10.1	.198	.184
FORUMS	3.8±2.3	14.2±12.7	.181	.171
BLOGS	4.1±2.8	18.5±24.8	.206	.203
WIKIPEDIA	4.5±2.8	21.9±16.2	.190	.188
BNC	4.3±2.8	19.8±14.5	.169	.168

Table 3: Average word and sentence length, and proportion of OOV words (optionally with lexical normalisation) in each dataset

spellings/social media usages such as *ur* for *your*, we optionally include a pre-step of “lexical normalisation” based on the dictionary of Han et al. (2012a) which gives the standard form for a given OOV, based on combined information from slang dictionaries and automatically-learned correspondences (“+norm”).

There is remarkably little difference in word length between datasets, but sentence length in TWITTER-1/2 and COMMENTS is around half that of the more formal WIKIPEDIA/BNC and also BLOGS, with FORUMS splitting the difference. The average word length for all of TWITTER-1/2, COMMENTS and FORUMS is remarkably similar. In terms of OOV words, FORUMS and COMMENTS are comparable to WIKIPEDIA and the BNC (where OOV words are dominated by proper nouns), and actually lower than BLOGS. TWITTER-1/2 has the highest OOV rate of all our datasets, although when we include lexical normalisation, it is only 2–4 percentage points higher than the other social media sources. The impact of lexical normalisation is most noticeable for TWITTER-1/2 and COMMENTS, indicating that informal text and “ad hoc” spellings are more prevalent in them than the other data sources.

about one third; it also reduced the OOV rate in COMMENTS by around 10%.

These results are broadly in agreement with the findings of Rello and Baeza-Yates (2012), who used the relative frequency of a set of common misspellings to estimate the lexical quality of social media, and arrived at the conclusion that social media text is on average “cleaner” than many other web sites, and becoming progressively cleaner over time.

### 5.3 Grammaticality

A natural next question to ask is how grammatical the text in each of our datasets is. We measure this using the English Resource Grammar (ERG: Flickinger et al. (2000)), a broad-coverage HPSG-based grammar. One aspect of the ERG which makes it highly suited to testing grammaticality is that, unlike most NLP parsers, it is “generative”, i.e. it explicitly models grammaticality, and is developed relative to both positive and negative test items to ensure it does not “overgenerate”. We can therefore use it as a proxy for grammaticality judgements. Further to this, the ERG makes active use of ‘root conditions’ to indicate how much the grammar had to relax particular assumptions to produce a derivation for the sentence. These conditions vary on the dimensions of: (1) strict versus informal (corresponding to whether the sentence uses standard punctuation and capitalisation, or not); and (2) full sentences vs. fragments (e.g. isolated noun phrases). All of our experiments are based on the ‘1111’ version of the grammar, and the CHEAP parsing engine (Callmeier, 2002).

In order to maximise the lexical coverage of the ERG, we used POS-conditioned generic lexical types (Adolphs et al., 2008), whereby a generic lexical entry is created for each OOV word on the basis of the output of a POS tagger. To accommodate the *TweetNLP* POS tags, we manually created a new set of mappings to generic lexical entries.<sup>8</sup> We additionally re-tokenised the output of *TweetNLP* to split apart contractions (e.g. *won’t* and possessive clitics (e.g. *Kim’s*), in line with the Penn Treebank tokenisation strategy.

In Table 4 we show the results of parsing 4000 randomly selected English sentences from each corpus using the ERG with the parsing setup we have described.<sup>9</sup>

The highest parse coverage was observed for

<sup>8</sup>The original POS mappings are based on the Penn POS tagset and have been tested and fine-tuned extensively; our POS mapping for the *TweetNLP* POS tags is much more immature, and has potentially contributed to a slight loss in

Corpus	Parseable				Unparseable
	strict		informal		
	full	frag	full	frag	
TWITTER-1	13.8	23.9	22.2	2.5	37.4
TWITTER-2	13.9	23.8	22.8	1.7	37.6
COMMENTS	18.0	22.2	26.4	1.4	31.9
FORUMS	23.9	14.1	24.7	1.5	35.6
BLOGS	25.6	17.5	18.8	2.7	35.3
WIKIPEDIA	48.7	4.5	18.9	1.5	26.2
BNC	38.4	12.0	24.0	2.2	23.2

Table 4: Percentage of sentences (from a random sample of 4000) which can be parsed using the ERG, broken down by the root condition of the top-ranked parse for the parseable sentences

the BNC (with only 23.2% not able to be parsed), closely followed by WIKIPEDIA. At the other end of the scale are the TWITTER-1 and TWITTER-2 variants, which are most likely to contain ungrammatical sentences, with up to 15% more sentences unable to be parsed, although this is only marginally higher than FORUMS and BLOGS, all of which contain more ungrammatical text than COMMENTS.

Between these extremes are some mild surprises — BLOGS and FORUMS, which contain data produced in a more enduring and editable format than TWITTER-1/2, are, according to our metric, only marginally more grammatical. In addition, the non-editable and relatively transient COMMENTS sentences are substantially more likely to be grammatical than either FORUMS or BLOGS. A large part of this effect however is probably due to the sentence length differences between the corpora. As shown in Table 3, the average length for COMMENTS is only 10.5 words, on par with TWITTER-1/2 (but according to this evidence, more carefully constructed). However, in the longer sentences of FORUMS and BLOGS, there is more scope for the authors to introduce anomalies into the text, increasing the chances of the sentence being unparseable.

Examining the root conditions related to formality and fragment analyses also gives us im-

parser accuracy relative to the “canonical” ERG.

<sup>9</sup>Note that the reported results differ significantly from the coverage numbers reported by Read et al. (2012b) for WIKIPEDIA in particular, through a combination of a generic sentence and word tokenisation strategy, a potentially lower-accuracy/coarser-grained POS tagger, and a less mature POS mapping. The impact of these factors should be constant across datasets, however, meaning that the relative numbers should be truly indicative of the relative grammaticality of their text content.

Corpus	Fragment	Preprocessor error	Resource limitations	Ungrammatical inputs	Extra-grammatical	Grammar gaps
TWITTER-1	0.16	0.24	0.00	0.32	0.09	0.18
TWITTER-2	0.19	0.22	0.00	0.31	0.10	0.17
COMMENTS	0.13	0.32	0.00	0.31	0.04	0.20
FORUMS	0.05	0.31	0.01	0.36	0.03	0.24
BLOGS	0.09	0.22	0.11	0.11	0.22	0.25
WIKIPEDIA	0.08	0.11	0.10	0.06	0.06	0.59
BNC	0.15	0.05	0.15	0.04	0.05	0.56

Table 5: A breakdown of the causes of parser error in the unparseable sentences for each dataset

portant insights into the corpora. WIKIPEDIA has by far the highest percentage of sentences with a strict, non-fragment analysis, much higher (10.3%) than the BNC even. In the less-edited corpora, of those sentences which are able to be parsed, a much smaller percentage are strict or full analyses, with the strict fragment analyses being most prevalent in TWITTER-1/2 and informal full analyses dominating in COMMENTS and FORUMS.

The spread of grammaticality numbers is perhaps not as large as we might have expected. There are a few reasons for this. One important point is that the POS-tagging using a very coarse-grained tag set has inevitably led to very general lexical entries for handling unknown words (so we are not even sure of the person, number and tense associated with a verb). This means that it is possible that some of the sentences have been spuriously identified as grammatical, since the very general types for unknown words give the grammar great flexibility in fitting a parse tree to the sentence, even where it may not be appropriate. Secondly it is possible that this POS-tagging has led to an explosion in the number of candidate parse trees, which can paradoxically lead to a small decrease in coverage over longer sentences of WIKIPEDIA and the BNC due to the risk of exceeding the parser timeout or memory limit.

In line with Baldwin et al. (2005), it is possible to shed further light on the quality of the grammaticality judgements, and also stylistic differences between the different corpora by manually analysing the unparseable sentences according to the cause for parse failure, as being due to: (1) a syntactic fragment (not explicitly handled by the ERG; e.g. noun and verb phrase fragments such as *coming home ...*, or standalone expletives such as *wow!*); (2) a preprocessor error (e.g. in sentence tokenisation or POS tagging); (3) parser resource limitations (usually caused by the

grammar running out of edges in the chart, or timing out); (4) ungrammatical strings; (5) extragrammatical strings (where non-linguistic phenomena associated with the written presentation, such as bullets or HTML markup, interface unpredictably with the grammar); and (6) lexical and constructional gaps in the grammar. A breakdown of parse failure over a randomly-selected subset of 100 unparseable sentences from each of the datasets, carried out by the first author, is presented in Table 5.

It is clear that the proportion of ungrammatical sentences is an underestimate, especially in the case of WIKIPEDIA and the BNC, where more than half of the “failures” are attributable to lexical or constructional gaps in the grammar.<sup>10</sup> For TWITTER-1/2, COMMENTS and FORUMS, however, the proportion of grammar gaps and genuinely ungrammatical inputs, respectively, is roughly equivalent, suggesting that our original findings for these datasets are an underestimate of the actual proportion of ungrammaticality, but that the relative proportions are accurate.

An additional observation that can be made from Table 5 is that preprocessing is a common cause of parser failure, primarily in sentence tokenisation (with multiple sentences tokenised into one), and to a lesser extent in POS tagging, and also occasional errors in language identification (only observed in the TWITTER-1/2 data).

Reflecting back over the combined results for grammaticality, we can conclude that there is less syntactic “noise” in social media text than we may have thought, and that while there is no doubt that WIKIPEDIA and the BNC contain less ungrammatical text than the other datasets, the relative occurrence of syntactically “noisy” text in TWITTER-1/2, COMMENTS, FORUMS and

<sup>10</sup>Or, indeed, shortcomings in our POS mapping for unknown words, although again, the relative impact of this should be constant across datasets.

Corpus	Homogeneity
TWITTER-1	549
TWITTER-2	553
COMMENTS	613
FORUMS	570
BLOGS	716
WIKIPEDIA	575
BNC	542

Table 7: Corpus homogeneity using  $\chi^2$  (smaller values indicate greater self-similarity)

BLOGS is relatively constant.

There is partial concordance between these findings and those of Hu et al. (2013), who examined textual properties of Twitter messages relative to blog, email, chat and SMS data, and also a newspaper. They found that Twitter messages were more formal than chat and SMS messages, and more similar to email and blog text in composition, in making prevalent use of standard constructions and lexical items.

#### 5.4 Corpus Similarity

So far we have examined the datasets individually. Next, we investigate how intrinsically similar in style and content the different datasets are. One possible approach to this is via calculation of “corpus similarity” between datasets and homogeneity within a given dataset. In one of the very few studies of measuring corpus similarity and homogeneity, Kilgarriff (2001) introduced a method based on  $\chi^2$ , whereby we measure the similarity of two corpora as the  $\chi^2$  statistic over the 500 most frequent words in the union of the corpora. One limitation of Kilgarriff’s method is that it is only applicable to corpora of equal size. We therefore use the five 1M token sub-corpora of each corpus in these experiments. We measure the similarity of two corpora as the average pairwise  $\chi^2$  similarity between their sub-corpora. We measure the homogeneity (or self-similarity) of a corpus as the average pairwise similarity between sub-corpora of that corpus.

The homogeneity scores in Table 7 indicate that social media text exhibits greater lexical variation (as captured by the  $\chi^2$  measure), and hence is less homogenous, than conventional text types (i.e. the BNC). TWITTER-1 and TWITTER-2 are the most homogenous of the social media corpora, and only fractionally less homogeneous than the BNC. BLOGS are much more diverse than the other corpora.

Turning to corpus similarity (Table 6), there appears to be a roughly linear partial ordering in the relative similarity between the corpora: TWITTER-1/2  $\equiv$  COMMENTS < FORUMS < BLOGS < BNC < WIKIPEDIA (as in, TWITTER-1/2 is more similar to FORUMS than it is to BLOGS, but more similar to BLOGS than the BNC, etc.). This can be observed most clearly based on the similarities of each other corpus with TWITTER-1/2 and WIKIPEDIA, but the similarities for all corpus pairs are consistent with this ordering. TWITTER-1 and TWITTER-2 are unsurprisingly the most similar corpora, with very little difference between the two crawls, suggesting that despite the real-time nature of Twitter, it is reasonably homogenous across time. We further see relatively high similarity between TWITTER-1/2 and COMMENTS, COMMENTS and FORUMS, and FORUMS and BLOGS.

#### 5.5 Language Modelling

Language modelling provides an alternative to estimating corpus similarity, based on the perplexity of a dataset relative to language models (LMs) trained over other partitions from the same dataset, and also partitions from other datasets. We construct open-vocabulary trigram LMs with Good-Turing smoothing using SRILM (Stolcke, 2002).

For each corpus, we build 5 LMs, each trained on 4 of the available 1M word sub-corpora. We then use each model to compute the perplexity of the held-out sub-corpus from the same dataset, as well as all sub-corpora for each other dataset. The results are presented in Figure 1 in the form of a box plot over the 5 LMs for a given training corpus (although the variance between LMs is usually so slight that the “box” appears as a single point).

For each corpus, the lowest perplexity is obtained on the held-out data from the same corpus. Overall, these results agree with those for  $\chi^2$  similarity, namely that there is a continuous spectrum, with TWITTER-1/2 and WIKIPEDIA as the two extremes and COMMENTS, FORUMS, BLOGS and the BNC between them, in that order. Along this spectrum, COMMENTS, FORUMS and BLOGS form a cluster, as do the BNC and WIKIPEDIA.

Combining these results with those for  $\chi^2$  similarity, it would appear that FORUMS is the “median” dataset, which is most similar to each of the other datasets. The implication of this finding is that if a statistical model (e.g. for POS dis-

	TWITTER-1	TWITTER-2	COMMENTS	FORUMS	BLOGS	WIKIPEDIA
TWITTER-2	4.0	—	—	—	—	—
COMMENTS	63.7	62.4	—	—	—	—
FORUMS	91.8	90.6	62.3	—	—	—
BLOGS	115.8	119.1	128.4	61.7	—	—
WIKIPEDIA	347.8	360.0	351.4	280.2	157.7	—
BNC	251.8	258.8	245.2	164.1	78.7	92.5

Table 6: Pairwise corpus similarity ( $\times 10^3$ ) using  $\chi^2$

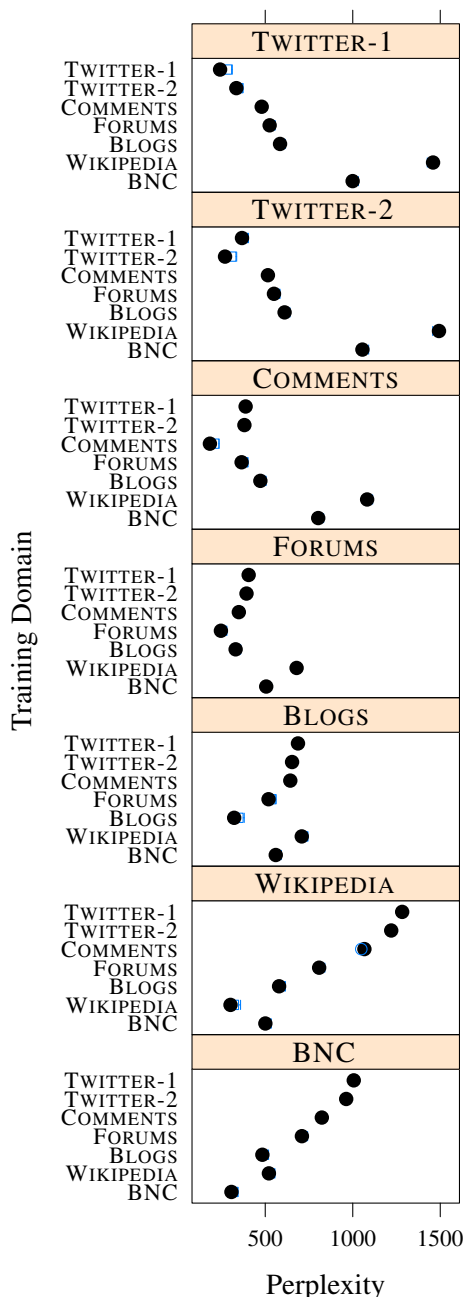


Figure 1: Trigram language model perplexity of test data conditioned on a given training corpus

ambiguation or parse selection) were to be trained on a single data type and applied to the other

data types, FORUMS should be the data of choice, as with the possible exception of WIKIPEDIA, it models the other corpora remarkably well. It also provides evidence for why methods based on edited text collections such as the BNC or newswire text perform badly on Twitter data.

## 6 Conclusions

In this paper we built corpora from a range of social media sources — microblogs, user-generated comments, user forums, blogs, and collaboratively-authored content — and compared them to each other and a reference corpus of more-conventional, edited documents. We applied a variety of linguistic and statistical analyses, specifically: language distribution, lexical analysis, grammaticality, and two measures of corpus similarity. This is the first such systematic analysis and cross-comparison of social media text.

We analysed the widely-acknowledged “noisiness” of social media texts from a number of perspectives, and showed that NLP techniques — including language identification, lexical normalisation, and part-of-speech tagging — can be applied to reduce this noise. Crucially, this suggests that although social media is indeed noisy, it appears to be possible to use NLP to “cleanse” it. Moreover, once rendered less noisy, (further) NLP on social media text might be more tractable than it is conventionally believed to be.

In terms of grammaticality, our results confirmed that social media text is less grammatical than edited text, but also suggested that the disparity is relatively small.

Both of our more-general corpus similarity analyses revealed that the social media text types analysed appear to lie on a continuum of similarity ranging from microblogs to collaboratively-authored content. This finding has potential implications on the selection of training data for statistical NLP systems.



## Acknowledgements

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT centre of Excellence programme.

## References

- Peter Adolphs, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Dan Flickinger, and Bernd Kiefer. 2008. Some fine points of hybrid natural language parsing. In *European Language Resources Association (ELRA), editor, Proc. of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 1380–1387, Marrakech, Morocco.
- Timothy Baldwin, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. 2005. Beauty and the beast: What running a broad-coverage precision grammar over the BNC taught us about the grammar — and the corpus. In *Stephan Kepsner and Marga Reis, editors, Linguistic Evidence: Empirical, Theoretical, and Computational Perspectives*, pages 49–69. Mouton de Gruyter, Berlin, Germany.
- Hila Becker, Mor Naaman, and Luis Gravano. 2009. Event identification in social media. In *Proceedings of the 12th International Workshop on the Web and Databases (WebDB 2009)*, Providence, USA.
- Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 389–398, Portland, USA.
- Lou Burnard. 2000. *User Reference Guide for the British National Corpus*. Technical report, Oxford University Computing Services.
- Kevin Burton, Niels Kasch, and Ian Soboroff. 2011. The ICWSM 2011 Spinn3r dataset. In *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM 2011)*, Barcelona, Spain.
- Ulrich Callmeier. 2002. PET – a platform for experimentation with efficient HPSG processing techniques. In *Stephan Oepen, Dan Flickinger, Jun’ichi Tsujii, and Hans Uszkoreit, editors, Collaborative Language Engineering*. CSLI Publications, Stanford, USA.
- Jacob Eisenstein, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287, Cambridge, MA.
- Jacob Eisenstein, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2012. Mapping the geographical diffusion of new words. *Arxiv preprint arXiv*, 1210.5268.
- 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 (NAACL HLT 2013)*, pages 359–369, Atlanta, USA.
- Dan Flickinger, Stephan Oepen, Hans Uszkoreit, and Jun’ichi Tsujii. 2000. On building a more efficient grammar by exploiting types. *Journal of Natural Language Engineering* (Special Issue on Efficient Processing with HPSG), 6(1):15–28.
- Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. WikiWoods: Syntacto-semantic annotation for English wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, Malta.
- Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proc. of the 5th International Joint Conference on Natural Language Processing*, pages 893–901, Chiang Mai, Thailand.
- Stephan Gouws, Dirk Hovy, and Donald Metzler. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90, Edinburgh, UK.
- 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 (ACL HLT 2011)*, pages 368–378, Portland, USA.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012a. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning 2012 (EMNLP-CoNLL 2012)*, pages 421–432, Jeju, Korea.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012b. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1045–1062, Mumbai, India.
- 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 (ACL 2013): System Demonstrations*, pages 7–12, Sofia, Bulgaria.
- Lichan Hong, Gregorio Convertino, and Ed H. Chi. 2011. Language matters in Twitter: A large scale study. In *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM 2011)*, Barcelona, Spain.
- Yuheng Hu, Kartik Talamadupula, and Subbarao Kambhampati. 2013. Dude, srsly?: The surprisingly formal nature of Twitters language. In *Proceedings of the 7th International Conference on Weblogs and Social Media (ICWSM 2013)*, Boston, USA.
- Akshay Java. 2007. A framework for modeling influence, opinions and structure in social media. In *Proceedings of the 22nd Annual Conference on Artificial Intelligence (AAAI-07)*, pages 1933–1934.
- Adam Kilgarriff. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.
- Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012) Demo Session*, pages 25–30, Jeju, Republic of Korea.
- Derek O’Callaghan, Martin Harrigan, Joe Carthy, and Pádraig Cunningham. 2012. Network analysis of recurring YouTube spam campaigns. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM 2012)*, pages 531–534, Dublin, Ireland.
- 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 (NAACL HLT 2013)*, Atlanta, USA.
- Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2012. Using paraphrases for improving first story detection in news and Twitter. In *Proc. of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–346, Montréal, Canada.
- Daniel Preotiuc-Pietro, Sina Samangooei, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjani. 2012. Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the ICWSM 2013 Workshop on Real-Time Analysis and Mining of Social Streams*, Dublin, Ireland.
- Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars JØrgen Solberg. 2012a. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India.
- Jonathon Read, Dan Flickinger, Rebecca Dridan, Stephan Oepen, and Lilja Øvrelid. 2012b. The WeSearch corpus, treebank, and treecache – a comprehensive sample of user-generated content. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1829–1835, Istanbul, Turkey.
- Luz Rello and Ricardo Baeza-Yates. 2012. Social media is NOT that bad! the lexical quality of social media. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM 2012)*, Dublin, Ireland.
- 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 (EMNLP 2011)*, pages 1524–1534, Edinburgh, UK.
- Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1104–1112, Beijing, China.
- Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldrige. 2012. Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning 2012 (EMNLP-CoNLL 2012)*, pages 1500–1510, Jeju Island, Korea.
- Semiocast. 2010. Half of messages on twitter are not in English — Japanese is the second most used language. Technical report, Semiocast.
- Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. of the International Conference on Spoken Language Processing*, volume 2, pages 901–904, Denver, USA.
- Benjamin Wing and Jason Baldrige. 2011. Simple supervised document geolocation with geodesic grids. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 955–964, Portland, USA.
- Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, 27(6):52–59.

# Who Had the Upper Hand? Ranking Participants of Interactions Based on Their Relative Power

**Vinodkumar Prabhakaran**  
Columbia University  
New York, NY, USA  
vinod@cs.columbia.edu

**Ajita John**  
Avaya Labs  
Basking Ridge, NJ, USA  
ajita@avaya.com

**Dorée D. Seligmann**  
Avaya Labs  
Basking Ridge, NJ, USA  
doree@avaya.com

## Abstract

In this paper, we present an automatic system to rank participants of an interaction in terms of their relative power. We find several linguistic and structural features to be effective in predicting these rankings. We conduct our study in the domain of political debates, specifically the 2012 Republican presidential primary debates. Our dataset includes textual transcripts of 20 debates with 4-9 candidates as participants per debate. We model the power index of each candidate in terms of their relative poll standings in the state and national polls. We find that the candidates' power indices affect the way they interact with others and the way others interact with them. We obtained encouraging results in our experiments and we expect these findings to carry across to other genres of multi-party conversations.

## 1 Introduction

Recently, there has been a rapid increase in social interactions being stored on the World Wide Web. In addition to those interactions that are inherently online such as discussion forums and social networks, offline interactions such as broadcast events, debates and speeches are also captured in real time and stored online in repositories such as YouTube and news media outlets. This growing mass of public data representing various modes of interactions enables researchers to computationally analyze social interactions at a scale which was not feasible previously. Within the field of analyzing online social interactions, there is a growing interest to study how the power or status difference between participants is reflected in

the various facets of interactions and if it can be detected using computational means (Diesner and Carley, 2005; Rowe et al., 2007; Bakshy et al., 2011; Bramsen et al., 2011; Biran et al., 2012; Danescu-Niculescu-Mizil et al., 2012).

When people interact with one another, there is often a power differential that affects the way they interact. This differential may be drawn from a multitude of factors such as social status, authority, experience, age etc. Identifying the dominant participants of an interaction through a power ranking system could have various applications. It could help improve effectiveness of advertisements within online communities. For example, targeting an advertisement to powerful and influential members within an online community might increase its effectiveness and reach to the community members. Power analysis can also help in information retrieval systems. Revealing power dynamics within stored interactions could be useful in determining relevance for a user with information needs. For example, a user may want to limit his search to posts authored by interactants with higher power. Power analysis may also aid intelligence agencies to detect leaders and influencers in suspicious online communities. This is especially useful since the real identities of the members of such communities are often not revealed and the hierarchies of such communities may not be available to the intelligence agencies.

Most computational efforts to analyze or predict power differentials between participants of interactions have relied on static power structures or hierarchies as sources for the power differential (Rowe et al., 2007; Bramsen et al., 2011; Gilbert, 2012). However, many interactions happen outside the context of a pre-defined static power structure or hierarchy. Examples for such interactions

include political debates, online discussions, and email interactions outside organizational boundaries. Although the participants of these interactions may not be part of an established power structure, there is often a power differential between them drawn from various other factors such as popularity, experience, knowledge etc. In such situations, the interaction itself plays an important role as a medium for the interactants to pursue, gain and maintain power over others. Consequently, the manifestations of power in such interactions will also inherently differ from the cases where a hierarchy is present. However, most computational studies on power within interactions have not explored such a dynamic notion of power.

In this paper, we analyze political debates where the power differential is dynamic. Specifically, we analyze the 2012 Republican presidential primary debates. We present an automatic ranking system to rank debate participants in terms of their relative power. We model the power of each candidate in terms of their relative standings in the polls released prior to the debate. We find that the candidates' power indices affect the way they interact with others and the way others interact with them. To our knowledge, our work is the first to do an in-depth computational analysis of the structure of interactions, modeling patterns of interruptions and mentions of participants, in relation to power. Moreover, the domain we study is particularly interesting since the primary objective of the debate participants is to pursue and maintain power over each other, as opposed to operating within a static power structure. Lastly, the findings of this study are note-worthy as they relate to the domain of political debates, an area which has not been well-studied in this fashion before. We will release the dataset with annotations to the research community to drive more research in this direction.

Next, we review the background and related computational work in the area of power analysis. Section 3 presents the domain of presidential debates and details how we model power in this domain. Section 4 presents the data and Section 5 presents the power ranker and describes features, experiments and results.

## 2 Background and Related Work

Social power and how it affects the ways people behave in interactions have been studied extensively in social sciences and psychology. Bales

and Slater (1955) studied interactions in small group conversations and suggested language as a reflection and resource of power and influence. Later, Bales (1970) identified the importance of the structure of conversations (e.g. frequency of turns) and argued that "to take up time speaking in a small group is to exercise power over the other members for at least the duration of the time taken, regardless of the content". Ng et al. (1993) found that conversational turns gained by interruptions are stronger indicators of power than turns gained otherwise. In further work Ng and Bradac (1993), they argued that the content also plays a role in influence; a view contrary to (Bales, 1970). More indicators of power in the content of interactions were studied later on. Sexton and Helmreich (1999) found linguistic indicators that could help identify relative status between individuals in social interactions. Locher (2004) studies politeness in interactions in relation to the exercise of power. Our work draws inspiration from many of these studies and looks for correlates of power in both the content and structure of interactions.

Due to the easy availability of data, most of these studies have been performed on written interactions, whereas our study is done on spoken interactions. Early computational approaches to analyze power in interactions relied on network-based approaches. There have been several studies using Social Network Analysis (Diesner and Carley, 2005; Rowe et al., 2007) for extracting social relations from emails. These approaches rely on collections of interactions between a set of individuals to build interaction networks and use various centrality metrics on those networks in order to deduce power relations between interactants. These studies mainly use the meta-data about messages: who sent how many messages to whom and when. Researchers have also analyzed the content of messages using NLP techniques to detect power differentials. For example, Bramsen et al. (2011) and Gilbert (2012) utilize a text classification approach and classify messages in the Enron email corpus as messages sent from a superior to a subordinate, and *vice versa*. Both studies model static hierarchical relationships; our work models a dynamic notion of power in interactions happening outside organizational boundaries. Also, the studies described above consider messages or collections of messages in isolation, but not in the context of the entire interaction.

More recently, a deeper analysis of interactions is shown to be useful in detecting power or influence in interactions. Danescu-Niculescu-Mizil et al. (2012) focus on the notion of language coordination — a metric that measures the extent to which a discourse participant adopts another’s language — in relation to various social attributes such as power, gender, etc. They perform their study on Wikipedia discussion forums and Supreme Court hearings — both of which have enforced power structures. Prabhakaran et al. (2012a) analyze the notion of overt displays of power (ODP) in dialog. Prabhakaran et al. (2012b) and Prabhakaran and Rambow (2013) study how the ODP and other dialog act analysis based features of organizational email interactions correlate with different types of power possessed by the participants. Biran et al. (2012) and Bracewell et al. (2012) use lower-level dialog constructs to model power relations. Biran et al. (2012) use dialog constructs such as attempts to persuade, agreement, disagreement and various dialog patterns in order to find influencers in Wikipedia discussion forums and LiveJournal blogs. Bracewell et al. (2012) try to identify participants pursuing power in discussion forums. They devise a set of eight *social acts* which largely overlaps with the dialog constructs used by (Biran et al., 2012).

Our work also falls into the above category of studies in the sense that we also go beyond pure lexical features and use dialog structure based features in our analysis. However, our work differs in few major ways. Firstly, our domain — political debates — contains spoken interactions while most studies discussed above are performed on written interactions (except Danescu-Niculescu-Mizil et al. (2012) which studies Supreme Court hearings). Secondly, in our domain, the primary purpose of the interactions is to pursue and maintain power, while most studies mentioned earlier deal with domains which are task oriented (Enron, Wikipedia and Supreme Court). Thirdly, in our domain, candidates may gain or lose power in the course of interactions, whereas power is more stable in the studies discussed above. Lastly, our interactions are time-bound, in contrast to online discussions such as Wikipedia forums.

We now turn our attention to related computational work on analyzing conversations in our domain of political debates. Rosenberg and Hirschberg (2009) analyze speeches made in the

context of 2004 Democratic presidential primary election and identify lexical and prosodic cues that signal charisma. More recently, Nguyen et al. (2012) analyze 2008 presidential and vice presidential debates to study how speaker identification helps topic segmentation and how candidates exercise control over conversations by shifting topics. While our domain is also presidential debates, our focus is on how the candidate’s power or confidence affects interactions within the debates.

### 3 Domain: Political Debates

Before the United States presidential election, a series of presidential primary elections are held in each U.S. state by both major political parties (Republican and Democratic) to select their respective presidential nominees. In recent times, it has become customary that candidates of both parties engage in a series of debates prior to and during their respective parties’ primary elections. In this study, we explore how the power differential between the candidates manifests in these debates. Specifically, we use the 20 debates held between May 2011 and February 2012 as part of the 2012 Republican presidential primaries.<sup>1</sup> There were a total of 10 candidates who took part in these primary debates; some of whom participated only in one or two debates. Interactions in these debates are fairly well structured and follow a pattern of the moderator asking questions and the candidates responding, with some disruptions due to interruptions from other candidates.

Presidential debates serve an important role during the election process. It serves as a platform for candidates to discuss their stances on policy issues and contrast them with other candidates’ stances. In addition, it also serves as a medium for the candidates to pursue and maintain power over other candidates. This makes it an interesting domain to investigate how power dynamics between participants are manifested in an interaction. In addition, the 2012 Republican presidential election campaign was one of the most volatile ones in recent times. Most candidates held the front runner position at some point during the campaign. This prevents the analysis of power dynamics in these debates from being biased on the personal characteristics of a single candidate or a small set of can-

---

<sup>1</sup>There were no Democratic presidential primary debates in 2012, since the incumbent President Barack Obama was the de-facto nominee.

didates. Figure 2 shows the trend of how power indices of candidates (to be defined formally in Section 3.1) varied across debates.

### 3.1 Modeling Power in Debates

We use the term *Power Index* to denote the power or confidence with which a candidate comes into the debate. The Power Index of a candidate can be influenced by various factors. For example, during the presidential primary election campaigns, candidates get endorsed by various political personalities, newspapers and businesses. We think that such endorsements as well as the funds raised through campaigns positively affect the Power Index of the candidate. However, a more important source of a candidate’s power is their relative standing in recent poll scores. It gives the candidate a sense of how successful he/she is in convincing the electorate of his/her candidature. In this study, we model the Power Index of each candidate based solely on their recent state or national poll standings because we think that this is the most dominant factor. Other components such as the funds raised can be included in a similar fashion in the calculation of Power Index. We leave this to future work.

For each debate  $D$ , we denote the set of candidates participating in that debate by  $C_D$ . Let  $date(D)$  denote the date on which debate  $D$  was held and  $state(D)$  denote the state in which it was held. Debates from December 2011 onwards were held in states where the primaries were to be held in the near future. In these debates, we assume that their standings in the respective state polls, rather than national polls, would be the dominating factor affecting the power or confidence of candidates. Hence, for those debates, we chose the respective state’s poll scores as the reference. For others, we chose the national polls as the reference. Let  $refType$  denote the type of the reference poll we consider for debate  $D$ .

$$refType = \begin{cases} state(D), & \text{if } date(D) > 12/01/11 \\ NAT, & \text{otherwise} \end{cases}$$

We show the  $refType$  for each debate in Figure 1. For each debate, we find the poll results (national or state) released most recently and use the percentage of electorate supporting each candidate as the power index. If there are multiple polls released on the day the most recent poll was released, then we take the mean of poll scores

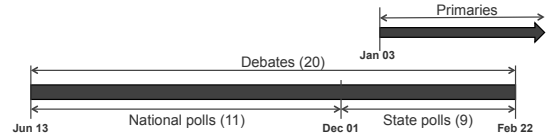


Figure 1: Timeline of Debates and Primaries

Number of debates	20
Interaction time	30-40 hrs
Average number of Candidates per debate	6.6
Average number of Turns per debate	245.2
Average number of Words per debate	20466.6

Table 1: Debate statistics

from all those polls to find the power index. Let  $RefPolls(D)$  be the set of polls of type  $refType$  released on the most recent date on which one or more such polls were released before  $Date(D)$ . We define the *Power Index*,  $\mathcal{P}(X)$ , of candidate  $X \in C_D$  as below

$$\mathcal{P}(X) = \frac{1}{|RefPolls(D)|} \sum_{i=1}^{|RefPolls(D)|} p_i$$

where  $p_i$  denote the poll percentage  $X$  got in the  $i^{th}$  poll in  $RefPolls(D)$ .

## 4 Data

We obtained the manual transcripts of presidential debates from The American Presidency Project.<sup>2</sup> The transcripts of all debates follow similar formats, except for a few exceptions. Each debate’s transcript lists the presidential candidates who participated and the moderator(s) of the debate. Transcripts demarcate speaker turns and also contain markups to denote applause, laughter, booing and crosstalk during the debates. Table 1 shows various statistics on the debates. We obtained the state and national poll results from the corresponding Wikipedia pages which kept track of polls from various sources including Gallup, various national and regional news agencies etc.<sup>3,4</sup> Figure 2 shows the trend of how the power indices of candidates varied across debates. Of the ten candidates, seven of them (everyone except Johnson, Huntsman and Pawlenty) were among the top 3 candidates for at least three debates.

<sup>2</sup><http://www.presidency.ucsb.edu/debates.php>

<sup>3</sup>[http://en.wikipedia.org/wiki/Statewide\\_opinion\\_polling\\_for\\_the\\_Republican\\_Party\\_presidential\\_primaries,\\_2012](http://en.wikipedia.org/wiki/Statewide_opinion_polling_for_the_Republican_Party_presidential_primaries,_2012)

<sup>4</sup>[http://en.wikipedia.org/wiki/Nationwide\\_opinion\\_polling\\_for\\_the\\_Republican\\_Party\\_2012\\_presidential\\_primaries](http://en.wikipedia.org/wiki/Nationwide_opinion_polling_for_the_Republican_Party_2012_presidential_primaries)

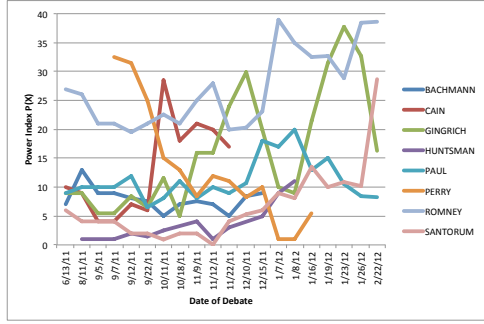


Figure 2: Power Index  $P(X)$  variations across debates  
 Note: Plots for Pawlenty and Johnson are not shown since they participated only in one or two debates.

## 5 Automatic Power Ranker

In this section, we present a supervised learning system to rank the participants of the debates based on their power indices. Formally, given a debate  $D$  with a set of participants  $C_D = \{X_1, X_2, \dots, X_n\}$  and corresponding power indices denoted by  $P(X_i)$  for  $1 < i < n$ , we want to find a ranking function  $r : C_D \rightarrow \{1 \dots n\}$  such that for all  $1 < i, j < n$ ,

$$r(X_i) > r(X_j) \iff P(X_i) > P(X_j)$$

We use an SVM based supervised learning system to estimate the ranking function  $r'$  that gives an ordering of participants  $\{X'_1, X'_2, \dots, X'_n\}$ , optimizing on the number of inversions between the orderings produced by  $r'$  and  $r$ .

### 5.1 Features

One of the primary ways power is manifested in an interaction is the manner in which people participate. By this, we are referring to the conscious and subconscious choices a participant makes while engaging in interactions. These include the lexical choices of each participant as well as other choices that affect the structure of the interaction - such as how much a participant speaks and on what topics. We used features to capture the language used in the debates as well as the structure of debates. Specifically, we analyze each debate participant in 4 dimensions — what they said (lexical features), how much they spoke (verbosity features), how they argued (argument features), and how they were talked about (mention features). Some structural features such as turns information are readily available from the transcripts, while for some others like arguments and candidate mentions, we use simple heuristics or perform deeper NLP analysis. The features we used are described in detail below

and are summarized in Table 2.

Code	Feature Description
Lexical: What they spoke	
WN	WordNgrams: word sequence of length 1 to 5
PN	PosNgrams: POS sequence of length 1 to 5
Verbosity: How much they spoke	
WD	WordDev: % of words spoken by $X - 1/ C_D $
TD	TurnDev: % of turns by $X - 1/ C_D $
QD	QuestionDev: % of questions to $X - 1/ C_D $
LT	LongestTurn: # of words in the longest turn
WT	WordsPerTurn: average # of words per turn
WS	WordsPerSent: average # of words per sentence
Argument: How they argued	
IOT	InterruptOthersPerTurn: % of candidate $X$ 's turns that were interrupting others
OIT	OthersInterruptPerTurn: % of candidate $X$ 's turns that others interrupted
Mentions: How they were talked about	
MP	MentionPercent: % of candidate $X$ mentions
FN	FirstNamePercent: % of candidate $X$ mentions that were first name mentions
LN	LastNamePercent: % of candidate $X$ mentions that were last name mentions
FLN	FirstAndLastNamePercent: % of candidate $X$ mentions that were first and last name mentions
TN	TitleAndNamePercent: % of candidate $X$ mentions that were mentions using titles

Table 2: Features with respect to candidate  $X$

**Lexical - What they said:** Ngram based features have been used in previous studies to analyze power in written interactions (Bramsen et al., 2011; Gilbert, 2012). It is expected to capture lexical patterns that denote power relations. We aggregated all turns of a participant and extracted counts for word lemma ngrams (WN) and POS tag ngrams (PN).

**Verbosity - How much they spoke:** We used features to capture each candidate's proportion of turns, time duration they talked, and number of questions posed to them. We approximated the time duration each speaker spoke by the total number of words spoken by him/her in the entire debate. To find the number of questions asked, we used the heuristic — instances where the candidate spoke right after the moderator are questions the moderator posed to the candidate. The percentage values of these features are dependent on the number of participants in each debate, which varied from 9 to 4. To handle this, for each feature, we measured the deviation of each candidate's percentage for that feature from its expected fair share percentage in the debate. We define the fair share percentage of a feature in a given debate to be  $1/|C_D|$  — the percentage each candidate would

```

...
SANTORUM: ... I would ask Governor Romney, do
you believe people who have -- who were felons,
who served their time, who have extended --
exhausted their parole and probation, should
they be given the right to vote?

WILLIAMS: Governor Romney?

ROMNEY: First of all, as you know, the PACs
that run ads on various candidates, as we
unfortunately know in this --

SANTORUM: I'm looking for a question -- an
answer to the question first. [applause]

ROMNEY: We have plenty of time. I'll get there.
I'll do it in the order I want to do. [...]
the super PACs run ads. [...] they said that
you voted to make felons vote? Is that it?

SANTORUM: That's correct. That's what the ad
says.

ROMNEY: And you're saying that you didn't?

SANTORUM: Well, first, I'm asking you to answer
the question, because that's how you got the
time. It's actually my time. [...] should
they be given the right to have a vote?

```

Figure 3: Excerpt from the debate held at Myrtle Beach, SC on January 16 2012

have gotten for that feature if it was equally distributed. We calculate the deviation of each feature — TurnDev (TD), WordDev (WD) and QuestionDev (QD) — as the difference between observed percentage for that feature and  $1/|C_D|$ . We also investigated three additional structural features - longest turn length (LT), words per turn (WT) — whether they had longer turns on average, and words per sentence (WS) — whether they used shorter sentences.

**Argument - How they argued:** Modeling arguments and interruptions in interactions is not a straight-forward task. There has been work in the NLP community to detect arguments and interruptions in spoken as well as written interactions (Somasundaran et al., 2007). However, the well-structured nature of interactions that is expected in the debates allows us to use some simple heuristics to detect arguments and interruptions for the purposes of this study. We leave deeper NLP processing of candidate turns to detect interruptions and arguments for future work.

Debates follow a pattern where the candidate is expected to speak only after a moderator prompts him or her to either answer a question or to respond to another candidate. Hence, if a candidate talks immediately after another candidate, he is disrupting the expected pattern of the debate. This holds true even if such an out-of-turn talk may

not have interrupted the previous speaker mid-sentence. We considered such instances where the candidate spoke out-of-turn after another candidate as interruptions to the previous candidate. In most cases, such interruptions lead to back-and-forth exchanges between the candidates until a moderator steps in. We define such exchanges between candidates where they talk with one another without the moderator intervening as an argument. Arguments can extend to many number of turns. In counting interruptions, we counted only the first interruption by each candidate in the series of turns that constitute an argument. An example argument is given in Figure 3 where we counted only one instance of interruption for both Santorum and Romney. We used features to capture interruptions by candidate  $X$  as well as interruptions by others while candidate  $X$  was speaking. Since the raw counts of these measures are dependent on the number of turns, we used the normalized counts to find the *per-turn* value of these measures as features — InterruptOthersPerTurn (IOT) and OthersInterruptPerTurn (OIT).

**Mentions - How they were talked about:** Intuitively, how often a candidate was mentioned or referred to by others in the debate is a good indicator of his or her power. The more a candidate is mentioned, the more central he or she is in the context of that debate. We use the mention count normalized across the total number of mentions of all candidates in a given debate (MP) as a feature.

In addition, we look at the form of addressing used while referring to each candidate. Previous studies in social sciences and linguistics have looked at the form of addressing in relation to the social relations (Brown and Ford, 1961; Dickey, 1997). Building on insights from these studies, we investigated if the modes of addressing candidates change with respect to their power. Specifically, we looked at four modes of addressing — FN (First Name), LN (Last Name) FLN (First and Last Name) and TN (Title followed by Name, first, last or full). As titles, we included common titles such as Mr., Ms. etc. as well as a set of domain-specific titles: Governor, Speaker, Senator, Congresswoman and Congressman. About 68.6% of total candidate mentions across debates were TN mentions, while the other types of mentions accounted for close to 10% each. FN, LN, TN and FLN capture the distribution of each candidate's mentions across these four types of mentions as



percentage of their total mentions.

## 5.2 Correlation Analysis and Significance

Figure 4 shows the Pearson’s product correlation between each structural feature and candidate’s power index  $P(X)$ . The darker bars denote statistically significant ( $p < 0.05$ ) correlations. Applying Bonferroni correction for multiple tests, the threshold for p-value for significance would be reduced to 0.0025. Even then, the statistically significant features would retain their significance. We consider three correlation windows — weak (0.2 - 0.39), moderate (0.4 - 0.69) and high (0.7 and above).

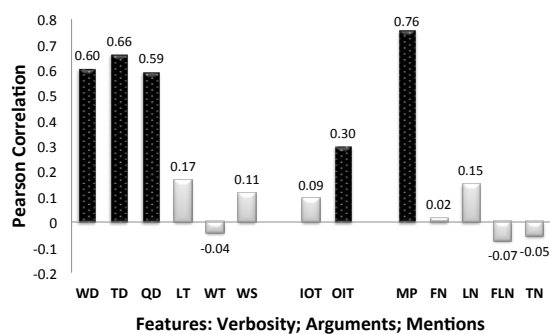


Figure 4: Pearson Correlations for Structural Features  
Correlation windows: Weak (0.2 - 0.39); Moderate (0.4 - 0.69); High ( $\geq 0.7$ )

We obtained statistically significant moderate positive correlation between the word and turn features and candidates’ power indices. Candidates with higher power indices spoke for significantly more time than others (WD) and they also got significantly more number of turns (TD). This finding is in line with the empirical findings in sociology literature (Ng et al., 1993; Reid and Ng, 2000). We also obtained moderate positive correlation between questions posed to the candidate and his or her power index, which suggests that the candidates with higher power indices were asked significantly more questions by the moderators.

Another interesting observation was on the interruption patterns. We obtained no significant correlation between how powerful a candidate was and how often he/she interrupted others (IOT). Instead, we found statistically significant positive correlation (although weak) for OIT, which means that the candidates with more power were interrupted significantly more by others. This is counter-intuitive and in contrast with previous findings by (Ng et al., 1995) that those who interrupt are more influential or powerful. We believe

that this is a manifestation of the participants pursuing power over each other rather than operating within a static power structure.

We found statistically significant high positive correlation between the power indices of candidates and how often they were referenced/mentioned by others (MP). In other words, as candidates gain more power, they are referenced significantly more by others. However, the distribution of mentions of a candidate across different forms of addressing (FN, LN, TN, FLN) did not have any correlation with the power indices of the candidate. This suggests that while forms of addressing is found to be correlated with power relations by previous studies (Brown and Ford, 1961; Dickey, 1997), they are not affected by the short term variations of power as in our domain.

## 5.3 Implementation

To build the ranker, we used the ClearTk’s *SVM<sup>rank</sup>* (Joachims, 2006) wrapper package. We also used the ClearTk wrapper for the Stanford CoreNLP package to perform basic NLP analysis on the speaker turn texts. The basic steps we performed include - tokenization, sentence segmentation, parts-of-speech tagging, lemmatization and named entity tagging.

## 5.4 Evaluation

We report results on 5-fold cross validation. We report three commonly used evaluation metrics for ranking tasks — Kendall’s Tau,  $nDCG$  and  $nDCG_3$ . Kendall’s Tau measures the similarity between two rankings based on the number of rank inversions (discordant pairings) between original and predicted ranking.  $nDCG$  employs a normalized discounted cumulative gain method which penalizes the inversions happening in the top of the ranked list more than those happening in the bottom.  $nDCG_3$  focuses only on the top 3 candidates from each debate.  $nDCG$  based metrics are more suitable for our purposes since it provides a way to factor in the magnitude of ranking metric (in our case, power index) in the performance assessment. E.g., under  $nDCG$ , the penalty for swapping a pair of candidates with  $P(X)$  values 35.0 and 5.0 will be higher than that for a pair with  $P(X)$  values 12.0 and 15.0. Tau treats these mistakes equally if the swaps generate the same number of inversions.



## 5.5 Results and Discussion

We first find the best performing set of lexical features (Word and POS ngrams) by varying the ngram length from 1 to 5. We then find the best performing feature subset of structural features among all subsets. The small cardinality of the set of structural features makes this feasible. We then use the combination of the best feature subsets from both settings. The results obtained are presented in Table 3. We present a baseline system using word unigrams as features.

	<b>Tau</b>	<b>nDCG</b>	<b>nDCG-3</b>
<i>Baseline (Unigrams)</i>	0.25	0.860	0.733
<i>WN+PN</i>	0.36	0.880	0.779
<i>WD+QD+MP</i>	<b>0.47</b>	<b>0.961</b>	<b>0.921</b>
<i>WD+QD+OIT</i>	0.45	0.960	<b>0.921</b>
<i>WN+PN+WD+QD+MP</i>	0.37	0.902	0.818
<i>WN+PN+WD+QD+OIT</i>	0.37	0.902	0.826

Table 3: Ranker results

We obtain the best configuration of lexical features to be WN+PN, with values of  $n$  as 1 and 2 respectively. The PN features improve the performance of the baseline system (unigrams) from 0.25 to 0.36  $Tau$ . Similar improvements are observed in  $nDCG$  and  $nDCG_3$  as well. The structural features outperform the lexical features and obtain the best overall result of 0.961 for  $nDCG$  and 0.921 for  $nDCG_3$  for a combination of WordDeviation, QuestionDeviation and MentionPercent. Another feature subset — WordDeviation, QuestionDeviation and OthersInterruptPerTurn — obtained the same performance in  $nDCG_3$ , but slightly lower numbers for  $Tau$  and  $nDCG$ . The overall best performing features were WD, QD, MP and OIT, which is in line with the findings in the correlation study in Section 5.2. WD suggests that people with more power tend to and/or are allowed to talk more. QD, MP and OIT are reflections of how others’ perception of a candidates power affected the way they interacted with him/her. Surprisingly, combining lexical and structural features did not yield good results. We suspect that this might be due to the high dimensional ngram feature space.

We analyzed the correlation of each structural features with  $P(X)$  in Section 5.2. However, it is not feasible to perform such significance studies on ngram features because of the huge feature space. In order to find the ngram features that are most representative for this task, we inspected

the feature weights of the linear kernel model created for the best performing ngram feature set (WN+PN). Table 4 lists few of the interesting features that came in the top 25 positive and negative weighted features, along with corresponding weights. POS tags are capitalized and `_BOS_` stands for `beginning_of_sentence`. It is hard to infer strong conclusions based purely on the SVM feature weights. However, SVM does pick up some interesting signals. E.g., those with power used `you` more, while those with less power used `we` more. Also, those with power used `agree` more, suggesting that they might be less contentious than others. `UH_`, which captures interjections/pauses was assigned a positive weight, which aligns with the finding that those with power get interrupted more. `_BOS__JJ` (-0.11) suggests that the participant with lower power tend to start sentences using an adjective.

Positive weighted	Negative weighted
<i>VBN_NN (0.30)</i>	<i>tell (-0.24)</i>
<i>agree (0.27)</i>	<i>do (-0.23)</i>
<i>UH_ (.18)</i>	<i>WDT (-0.15)</i>
<i>you (0.09)</i>	<i>we (-0.09)</i>
<i>VBP_TO (0.18)</i>	<i>_BOS__JJ (-0.11)</i>

Table 4: Top weighted features from the ngram based model created for WN + PN

## 6 Conclusion and Future Work

We presented a system to automatically rank participants of an interaction in terms of their relative power. We identified several linguistic and structural features that were effective in predicting these rankings. We conducted this study in the domain of political debates, specifically the 2012 Republican presidential primary debates. We find that candidates’ power indices affected the way they interacted with others in the debates — how much they spoke and how they spoke. We also found that power affected the way others interacted with them — the number of questions directed at them, how often they were interrupted, and how often they were mentioned. Our experiments in this domain yield very encouraging results and we plan to investigate if these findings carry across to other genres of multi-party conversations as a part of our future work. We also plan to perform deeper analysis on the interactions such as looking for dialog patterns which may signal topic control in relation to power.

## References

- E. Bakshy, J.M. Hofman, W.A. Mason, and D.J. Watts. 2011. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM.
- R.F. Bales and P.E. Slater. 1955. Role differentiation in small decision-making groups. *Family, socialization and interaction process*, pages 259–306.
- R. F. Bales. 1970. *Personality and interpersonal behaviour*. New York: Holt, Reinhart, and Winston.
- O. Biran, S. Rosenthal, J. Andreas, K. McKeown, and O. Rambow. 2012. Detecting influencers in written online conversations. In *Proceedings of the Second Workshop on Language in Social Media*, pages 37–45, Montréal, Canada, June. ACL.
- D. B. Bracewell, M. Tomlinson, and H. Wang. 2012. A motif approach for identifying pursuits of power in social discourse. In *ICSC*, pages 1–8. IEEE Computer Society.
- P. Bramsen, M. Escobar-Molano, A. Patel, and R. Alonso. 2011. Extracting social power relationships from natural language. In *The 49th Annual Meeting of the ACL*, pages 773–782. ACL.
- R. Brown and M. Ford. 1961. Address in american english. *The Journal of Abnormal and Social Psychology*, 62(2):375.
- C. Danescu-Niculescu-Mizil, L. Lee, B. Pang, and J. Kleinberg. 2012. Echoes of power: language effects and power differences in social interaction. In *Proceedings of the 21st international conference on WWW*, New York, NY, USA. ACM.
- E. Dickey. 1997. Forms of address and terms of reference. *Journal of Linguistics*, pages 255–274.
- J. Diesner and K. M. Carley. 2005. Exploration of communication networks from the enron email corpus. In *In Proc. of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining 2005*, pages 21–23.
- E. Gilbert. 2012. Phrases that signal workplace hierarchy. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12*, pages 1037–1046, New York, NY, USA. ACM.
- T. Joachims. 2006. Training Linear SVMs in Linear Time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.
- M. A. Locher. 2004. *Power and politeness in action: disagreements in oral communication*. Language, power, and social process. M. de Gruyter.
- S. H. Ng and J. J. Bradac. 1993. *Power in language: Verbal communication and social influence*. Sage Publications, Inc.
- S. H. Ng, D. Bell, and M. Brooke. 1993. Gaining turns and achieving high in influence ranking in small conversational groups. *British Journal of Social Psychology*, pages 32,265–275.
- S. H. Ng, M Brooke, , and M. Dunne. 1995. Interruption and in influence in discussion groups. *Journal of Language and Social Psychology*, pages 14(4),369–381.
- V. Nguyen, J. Boyd-Graber, and P. Resnik. 2012. SITS: A hierarchical nonparametric model using speaker identity for topic segmentation in multi-party. In *Association for Computational Linguistics*.
- V. Prabhakaran and O. Rambow. 2013. Written Dialog and Social Power: Manifestations of Different Types of Power in Dialog Behavior. In *Proceedings of the IJCNLP*, Nagoya, Japan, October. ACL.
- V. Prabhakaran, O. Rambow, and M. Diab. 2012a. Predicting Overt Display of Power in Written Dialogs. In *Proceedings of the HLT-NAACL*, Montreal, Canada, June. ACL.
- V. Prabhakaran, O. Rambow, and M. Diab. 2012b. Who's (Really) the Boss? Perception of Situational Power in Written Interactions. In *Proceedings of the 24th International Conference on COLING*, Mumbai, India. ACL.
- S. A. Reid and S. H. Ng. 2000. Conversation as a resource for in influence: evidence for prototypical arguments and social identification processes. *European Journal of Social Psych.*, pages 30,83–100.
- A. Rosenberg and J. Hirschberg. 2009. Charisma perception from text and speech. *Speech Communication*, 51(7):640 – 655. Research Challenges in Speech Technology: A Special Issue in Honour of Rolf Carlson and Bjrn Granstrm.
- R. Rowe, G. Creamer, S. Hershkop, and S.J. Stolfo. 2007. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web Mining and Social Network Anal.* ACM.
- J. B. Sexton and R. L. Helmreich. 1999. Analyzing cockpit communication: The links between language, performance, error, and workload. In *Proceedings of the Tenth International Symposium on Aviation Psychology*, pages 689–695.
- S. Somasundaran, J. Ruppenhofer, and J. Wiebe. 2007. Detecting arguing and sentiment in meetings. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*.

# Readability Indices for Automatic Evaluation of Text Simplification Systems: A Feasibility Study for Spanish

**Sanja Štajner**

Research Group in Computational Linguistics  
University of Wolverhampton, UK  
sanjastajner@wlv.ac.uk

**Horacio Saggion**

TALN Research Group  
Universitat Pompeu Fabra, Spain  
horacio.saggion@upf.edu

## Abstract

This paper addresses the problem of automatic evaluation of text simplification systems for Spanish. We test whether already-existing readability formulae would be suitable for this task. We adapt three existing readability indices (two measuring lexical complexity and one measuring syntactic complexity) to be computed automatically, which are then applied to a corpus of original news texts and their manual simplifications aimed at people with cognitive disabilities. We show that there is a significant correlation between each of the three readability indices and several linguistically motivated features which might be seen as reading obstacles for various target populations. Furthermore, we show that there is a significant correlation between the two readability indices which measure lexical complexity.

## 1 Introduction

In recent years, there has been growing effort to simplify written material and make it equally accessible to everyone. Various studies indicate that lexically and syntactically complex texts can be very difficult for non-native speakers and people with various reading impairments (e.g. autistic, aphasic, dyslexic or deaf people). Aphasic people, for instance, may encounter problems with less frequent words and some particular sentence constructions (Devlin, 1999). They also have problems in understanding syntactic constructions which do not follow the canonical subject-verb-object structure (e.g. passive constructions), and especially those sentences which are semantically reversible, e.g. *“The boy was kissed by the girl”* (Carroll et al., 1999). Additionally, aphasic readers may have additional problems with

comprehending newswire texts which have some genre-specific characteristics. These types of texts tend to use long sentences, noun compounds and long sequences of adjectives, e.g. *“Twenty-five-year-old blonde-haired mother-of-two Jane Smith”* (Carroll et al., 1999). People with intellectual disabilities have problem with both lexically and syntactically complex texts, as well as with processing and loading large amounts of information (Feng, 2009).

Since the late nineties, several initiatives which proposed guidelines for producing plain, easy-to-read and more accessible documents have emerged, e.g. “The Plain Language Action and Information Network (PLAIN)”<sup>1</sup>, “Make it Simple, European Guidelines for the Production of Easy-to-Read Information for people with Learning Disability” (Freyhoff et al., 1998), “Am I making myself clear? Mencap’s guidelines for accessible writing”<sup>2</sup>, and “Web content accessibility guidelines”<sup>3</sup>. All these guidelines share similar instructions for accessible writing, some of them more detailed than others. They all advise the writer to use the active voice instead of passive, use the simplest form of a verb (present and not conditional or future), avoid hidden verbs (i.e. verbs converted into a noun), use short, simple words and omit unnecessary words, write short sentences and cover only one main idea per sentence, etc.

Armed with these guidelines and the aim of making written documents equally accessible to everyone, many attempts have been made to completely or at least partially automate the process of text simplification, which is very expensive and time-consuming when performed manually. So far, text simplification systems have been devel-

<sup>1</sup><http://www.plainlanguage.gov/>

<sup>2</sup><http://november5th.net/resources/Mencap/Making-Myself-Clear.pdf>

<sup>3</sup><http://www.w3.org/TR/WCAG20/>

oped for English (Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Wubben et al., 2012), Spanish (Saggion et al., 2011), and Portuguese (Aluísio et al., 2008), with recent attempts at Basque (Aranzabe et al., 2012), Swedish (Rybing et al., 2010), and Dutch (Ruiter et al., 2010). With the emergence of these systems, the question we are faced with is how to automatically evaluate their performance given that the access to the target users might be difficult.

This study is an attempt to address this issue. We focus on text simplification systems for Spanish and investigate whether some of the already existing readability indices could be used for the automatic evaluation of these systems. Using a corpus of original news texts and their manual simplifications which followed specific guidelines for writing for people with cognitive disabilities, we show that two lexical complexity indices – one suggested by Anula (2007), and other by Spaulding (1956) – are highly correlated in both these text sets. Furthermore, we show that both these indices and the third readability index concerned with syntactic complexity (Anula, 2007) could be used for automatic evaluation of text simplification systems, as each index is correlated with some subset of the linguistically motivated complexity features considered as obstacles for people with different reading impairments.

The remainder of the article is structured as follows: Section 2 presents the most important previous work on readability prediction and linguistically motivated complexity features considered as obstacles for people with different reading difficulties; Section 3 describes the corpora, features, and readability indices used in this study; Section 4 presents and discusses the results of analysis of three chosen readability indices, twelve linguistically motivated complexity features, and their mutual correlation; while Section 5 concludes the article by summarising the main contributions and proposing possible directions for future work.

## 2 Related Work

Since the 1950s, over 200 readability formulae have been developed (for the English language), with over 1000 studies of their application (DuBay, 2004). Initially, they were used to assess the grade level of textbooks. Later, they were adapted to different domains and purposes, e.g. to measure readability of technical manuals

(Automated Readability Index (Smith and Senter, 1967)), or US healthcare documents intended for the general public (the SMOG grading (McLaughlin, 1969)). Some of these first readability formulae are still widely in use, given their simplicity (they require only the average sentence and word length) and good correlation with the reading tests. One of the most used readability formulae – the Flesch Reading Ease score (Flesch, 1949) – for example, “correlates .70 with the 1925 McCall-Crabbs reading test and .64 with the 1950 version of the same test” (DuBay, 2004). Another set of readability formulae are those which depend on average sentence length and the percentage of words which cannot be found on a list of the “easiest” words, e.g. the Dale-Chall readability formulae (Dale and Chall, 1948). These formulae have been adapted to other languages by changing the coefficient before the factors (e.g. the Flesch-Douma (Douma, 1960) and Leesindex Brouwer (Brouwer, 1963) formulae for Dutch represent the adaptations of the Flesch Reading Ease score, while Spaulding’s Spanish readability formula (Spaulding, 1956) could be seen as an adaptation of the Dale-Chall formula (Dale and Chall, 1948)). Oosten et al. (2010) showed that readability formulae which are solely based on superficial text characteristics (average sentence and word length) seem to be strongly correlated even across different languages (English, Dutch, and Swedish).

With the recent advances of natural language processing (NLP) tools and techniques, new approaches to readability assessment have emerged. Schwarm and Ostendorf (2005), and Petersen and Ostendorf (2009), used statistical language modeling and support vector machines to show that more complex features (e.g. average height of the parse tree, average number of noun and verb phrases, etc.) give better readability prediction than the traditional Flesch-Kincaid readability formula. They based their approach on the texts from Weekly Reader<sup>4</sup>, and two smaller corpora: Encyclopedia Britannica and Britannica Elementary (Barzilay and Elhadad, 2003), and CNN news stories and their abridged versions<sup>5</sup>. Feng et al. (2009) introduced some new cognitively motivated features which should improve automatic readability assessment of texts for people with cognitive dis-

<sup>4</sup><http://www.weeklyreader.com/>

<sup>5</sup><http://literacynet.org/cnnsf/>

abilities. In addition to three previously used corpora (Weekly Reader, Britannica, and CNN news stories) aimed at second language learners or children, Feng et al. (2009) used a corpus of local news articles which were simplified by human editors in order to make them more accessible for people with mild intellectual disabilities (MID). The texts were further rated for readability by people with MID. The study (Feng et al., 2009) showed that their newly introduced cognitively motivated features (e.g. entity mentions, lexical chains, etc.) are better correlated with the user-study comprehension than the Flesch-Kincaid Grade Level index (Kincaid et al., 1975).

Štajner et al. (2012) stated that many features which could be automatically extracted from a parser’s output can indicate the occurrence of the obstacles to reading comprehension faced by people with autism. The authors referred to the syntactic concept of the projection principle (Chomsky, 1986) that “lexical structure must be represented categorically at every syntactic level” which implies “that the number of noun phrases in a sentence is proportional to the number of nouns in that sentence, the number of verbs in a sentence is related to the number of clauses and verb phrases, etc.” (Štajner et al., 2012). Therefore, they automatically extracted nine features which account for indicators of structural complexity (nouns, adjectives, determiners, adverbs, verbs, infinitive markers, coordinating conjunctions, subordinating conjunctions, and prepositions), and three which account for indicators of ambiguity in meaning (pronouns, definite descriptions, and word senses). Štajner et al. (2012) showed that many of these features are significantly correlated with the Flesch Reading Ease score (Flesch, 1949). Given that all of the reading obstacles for people with autism (Štajner et al., 2012) would also be difficult to understand for people with cognitive disabilities (Freyhoff et al., 1998; Feng, 2009), we believe that these features (Section 3.3) could also be a good measure of complexity reduction achieved in a text simplification system.

Motivated by the study of Štajner et al. (2012), we wanted to explore how these features are correlated with the existing readability formulae (this time for Spanish instead of English). These formulae were not initially intended to be used for the evaluation of text simplification systems but

rather to measure the grade level necessary to understand a given text. Therefore, we wanted to establish whether those readability indices could be used in an automatic evaluation of text simplification systems. To the best of our knowledge, this is the first study of this type for Spanish. Unlike the study of Štajner et al. (2012) which uses the Simple Wikipedia<sup>6</sup> as an example of simplified texts (which do not comply totally with easy-to-read guidelines for people with cognitive disabilities, but are rather intended for a much wider audience), our study uses the original news texts and their manual simplifications aimed at people with cognitive disabilities, following specifically tailored easy-to-read guidelines for this target population (Section 3).

### 3 Methodology

The corpora, readability indices and linguistically motivated complexity features used in this study are presented in the next three subsections.

#### 3.1 Corpora

We first compared all features and readability measures on a parallel corpus of original and manually simplified texts (Table 1) in order to investigate whether these complexity measures differ significantly on these two types of texts, thus justifying the idea to use them to measure the degree of the performed simplification. The corpus contains 200 original news articles in Spanish (provided by the Spanish news agency Servimedia<sup>7</sup>) and their manually simplified versions. Simplification was done by trained human editors, familiar with the particular needs of a person with cognitive disabilities and following a series of easy-to-read guidelines suggested by Anula (2007), as a part of the Simplext project<sup>8</sup> (Saggion et al., 2011).

Table 1: Corpora

Corpus	Texts	Sentences	Words
Original	200	1150	37121
Simplified	200	1804	24332

The simplification operations applied by human editors could be classified in the following four categories (Drndarevic et al., 2013):

<sup>6</sup><http://simple.wikipedia.org>

<sup>7</sup><http://www.servimedia.es>

<sup>8</sup><http://www.simplext.es/>

1. **Syntactic operations:** changes applied at the sentence level, such as sentence splitting or quotation inversion.
2. **Lexical operations:** infrequent, long or technical terms are substituted with their simpler synonyms, and certain expressions are paraphrased or otherwise modified.
3. **Content reduction:** a significant portion of original content is eliminated through summarisation and paraphrases, in accordance with the guidelines that indicate that only the most essential piece of information should be preserved.
4. **Clarification:** certain complex terms and concepts, for which no synonym can be found, are explained by means of a definition.

### 3.2 Readability Indices

In this study, we focused on three readability formulae for Spanish: two concerned with the lexical complexity of the text – LC (Anula, 2007) and SSR (Spaulding, 1956); and the third one concerned with the syntactic complexity of the given text – SCI (Anula, 2007).

**The Spaulding’s Spanish Readability index** (SSR) has been used for assessing the reading difficulty of fundamental education materials for Latin American adults of limited reading ability and for the evaluation of text passages of the foreign language tests (Spaulding, 1956). It predicts the relative difficulty of reading material based on the vocabulary and sentence structure, using the following formula:

$$SSR = 1.609 \times \frac{|w|}{|s|} + 331.8 \times \frac{|rw|}{|w|} + 22.0$$

Here,  $|w|$  and  $|s|$  denote the number of words and sentences in the text, while  $|rw|$  denotes the number of rare words in the text. According to Spaulding (1956), *rare words* are those words which cannot be found on the list of 1500 most common Spanish words provided in the same study<sup>9</sup>. Given that the SSR index was used for assessing the reading difficulty of the materials

<sup>9</sup>Detailed instructions on what should be considered as a *rare word* (e.g. special cases of numbers, names of months and days, proper and geographic names, initials, diminutives and augmentatives, etc.) can be found in (Spaulding, 1956). Here we do not apply rules (a)–(g) specified in (Spaulding, 1956).

aimed at adults of limited reading ability, it is reasonable to expect that this formula could be used for estimating the level of simplification performed by text simplification systems aimed at making texts more accessible for this target population.

**The Lexical Complexity index** (LC) was suggested by Anula (2007) as a measure of lexical complexity of literary texts aimed at the second language learners. It is calculated using the following formula:

$$LC = \frac{LDI + ILFW}{2}$$

where *LDI* and *ILFW* represent the *Lexical Density Index* and *Index of Low-Frequency Words*, respectively:

$$LDI = \frac{|dcw|}{|s|},$$

$$ILFW = \frac{|lfw|}{|cw|} \times 100$$

Here,  $|dcw|$ ,  $|s|$ ,  $|lfw|$ , and  $|cw|$  denote the number of distinct content words, sentences, low-frequency words, and content words (nouns, adjectives, verbs, and adverbs), respectively. Anula (2007) considers as *low frequency words* those words whose frequency rank in the Referential Corpus of Contemporary Spanish<sup>10</sup> is lower than 1,000.<sup>11</sup>

**The Sentence Complexity Index** (SCI) was proposed by Anula (2007) as a measure of sentence complexity in a literary text aimed at second language learners. It is calculated by the following formula:

$$SCI = \frac{ASL + ICS}{2}$$

where *ASL* denotes the average sentence length, and *ICS* denotes the index of complex sentences. They are calculated as follows:

$$ASL = \frac{|w|}{|s|},$$

$$ICS = \frac{|cs|}{|s|} \times 100$$

<sup>10</sup><http://corpus.rae.es/lfrecuencias.html>

<sup>11</sup>Both lists (from Referential Corpus of Contemporary Spanish and the Spaulding’s list of 1500 most common Spanish words) were lemmatised using Connexor’s parser in order to retrieve the frequency of the lemma and not a word form (action carried out manually in the two cited works).

Here,  $|w|$ ,  $|s|$ , and  $|cs|$  denote the number of words, sentences and complex sentences in the text, respectively.<sup>12</sup>

### 3.3 Linguistically Motivated Complexity Features

Inspired by the work of Štajner et al. (2012), and easy-to-read guidelines for writing for people with cognitive disabilities (Freyhoff et al., 1998), this study employs twelve linguistically motivated complexity features (Table 2). The first nine features (1–9) are indicators of structural complexity and the final three features (10–12) are indicators of ambiguity in meaning.

Table 2: Linguistically motivated features

#	Code	Feature
1	<i>N</i>	Noun
2	<i>Det</i>	Determiner
3	<i>Adj</i>	Adjective
4	<i>V</i>	Verb
5	<i>Inf</i>	Infinitive
6	<i>Adv</i>	Adverb
7	<i>Prep</i>	Preposition
8	<i>CC</i>	Coordinating conjunction
9	<i>CS</i>	Subordinating conjunction
10	<i>Pron</i>	Pronoun
11	<i>Sens</i>	Number of senses per word
12	<i>Amb</i>	Percentage of ambiguous words

The corpora were parsed with the Connexor’s Machine parser<sup>13</sup> and the features 1–10 (Table 2) were automatically extracted using the parser’s output. Features 11 and 12 were extracted using two lexical resources – the Spanish Open Thesaurus (version 2)<sup>14</sup> and the Spanish EuroWordNet (Vossen, 1998). The Spanish Open Thesaurus lists 21,831 target words (lemmas) and provides a list of word senses for each word. Each word sense is, in turn, a list of substitute words. There is a total of 44,353 such word senses. The Spanish part of EuroWordNet is far more exhaustive containing 50,526 word meanings and 23,370 synsets. For computation of measures related to word sentences we only considered the lemmas present in the lexical resources used. For each text we com-

<sup>12</sup>We consider a complex sentence one that contains multiple finite predicates according to the output of Connexor’s Machine parser.

<sup>13</sup>[www.connexor.eu](http://www.connexor.eu)

<sup>14</sup><http://openthes-es.berlios.de/>

pute the average number of senses per word (code *Sens*, Table 2) as well as the percentage of ambiguous words in the text (code *Amb*, Table 2) producing two measures for each lexical resource used (*SensWN*, *SensOT*, *AmbWN*, *AmbOT*, Section 4). In the computation we consider all occurrences of lemmas including repeated lemmas.

## 4 Results and Discussion

The results of the analysis of readability indices on the corpora and their mutual correlation are presented in Section 4.1, and the results of the analysis of linguistically motivated complexity features are presented in Section 4.2, while their correlation with the readability indices is presented and discussed in Section 4.3.

### 4.1 Analysis of Readability Indices

The results of the comparison of readability indices across the corpora are given in Table 3. Columns ‘Original’ and ‘Simple’ contain the mean value of the corresponding readability indices in each of the two corpora, while the column ‘Rel.diff.’ contain the mean value of the relative differences between the text pairs (original and simplified). Column ‘Sign.’ presents the level of significance at which the differences between the two corpora are statistically significant. For the indices which follow approximately normal distribution, this column contains the result of the paired t-test. For those which do not follow normal distribution, it contains the result of the alternative non-parametric test – the Wilcoxon signed-rank test. All tests of normality and statistical significance were performed in SPSS.

Table 3: Readability indices

Index	Original	Simple	Rel.diff.	Sign.
LC	21.05	12.76	-39.06%	0.001
SSR	184.20	123.82	-32.60%	0.001
SCI	41.36	29.99	-27.43%	0.001

The results presented in Table 3 clearly demonstrate that there is a significant difference between the original and manually simplified texts for all three readability indices. The text pairs show an average relative difference of almost 40% for LC and about 30% for SSR and SCI, thus justifying the idea that those readability indices might be used in an automatic evaluation of text simplifica-

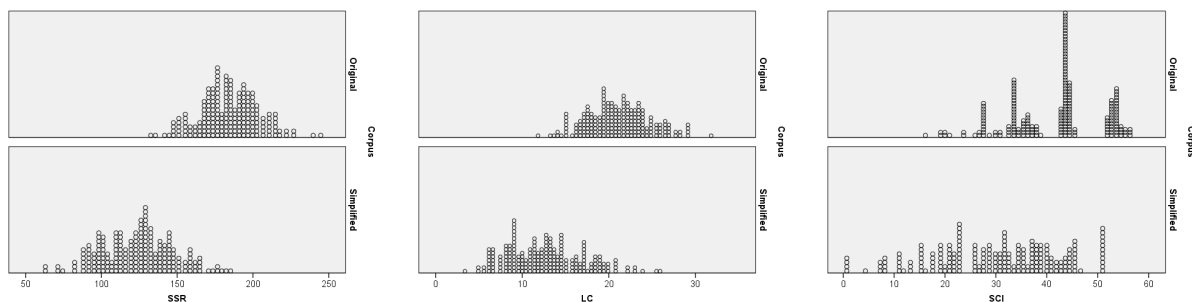


Figure 1: Readability indices across the corpora

tion systems as a measure of the degree of simplification. The distribution of the three readability indices (LC, SSR, and SCI) is presented in Figure 1, which shows that the distribution of all three indices is shifted left in the case of the simplified texts, thus indicating lower level of complexity.

The correlations between each pair of readability indices (LC–SSR, LC–SCI, and SSR–SCI), calculated using both corpora, are given in Table 4. All correlations which were reported as statistically significant at a 0.001 level of significance are presented in bold. As expected, the two readability indices concerned with the lexical complexity (LC and SSR) are significantly correlated, while the third one concerned with the syntactic complexity (SCI) is not significantly correlated with any of the other two (LC and SSR). The linear correlation between LC and SSR (measured by the Pearson’s coefficient) is, however, much less strong than the one among the four readability indices for English: Flesch Reading Ease, Flesch-Kincaid, Fog and SMOG, reported by Štajner et al. (2012).

Table 4: Correlation among readability indices

Corpus	Indices	Pearson	Spearman
Original	LC–SSR	<b>0.445</b>	<b>0.440</b>
	LC–SCI	-0.075	-0.085
	SSR–SCI	0.045	0.043
Simplified	LC–SSR	<b>0.353</b>	<b>0.378</b>
	LC–SCI	0.093	-0.116
	SSR–SCI	-0.159	-0.136

#### 4.2 Analysis of Linguistically Motivated Complexity Features

Occurrences of each feature which is an indicator of structural complexity, and prepositions (*Prep*)

were calculated as number of occurrences per 100 words. Average number of senses per word and percentage of ambiguous words in text were calculated in two different ways – using the Spanish EuroWordNet (SenseWN and AmbWN) and using the Spanish Open Thesaurus (SenseOT and AmbOT). The results of the analysis are presented in Table 5, using the same notation as in the case of readability indices in Table 3.

Table 5: Complexity Features

Feature	Original	Simple	Rel.diff.	Sign.
N	33.12	33.32	+1.13%	no
Det	14.82	17.13	+17.65%	0.001
Adj	7.24	4.89	-31.10%	0.001
V	10.39	14.56	+45.70%	0.001
Inf	1.65	2.22	+38.14%	0.001
Adv	2.27	3.35	+83.45%	0.001
Prep	19.75	17.12	-12.42%	0.001
CC	2.97	1.63	-41.79%	0.001
CS	1.82	2.55	+53.96%	0.001
Pron	19.75	17.12	+11.82%	0.001
SenseWN	3.78	4.01	+6.99%	0.001
AmbWN	66.02	72.19	+9.62%	0.001
SenseOT	3.52	3.65	+4.47%	0.001
AmbOT	78.89	82.71	+5.13%	0.001

The results in Table 5 show that the number of occurrences (per 100 words) of nouns does not differ significantly between the two corpora. Simplified texts have significantly lower number of occurrences (per 100 words) of adjectives, prepositions and coordinating conjunctions. This could be interpreted as an indication of omitting unnecessary information (adjectives), removing/resolving syntactic ambiguity and complexity (prepositions) and sentence splitting (coordinating conjunctions) in the process of simplification. The increased percentage of verbs might be a reflection of omitting



Table 6: Spearman’s correlation between readability indices and complexity features (Original)

Feature	LC	SSR	SCI
V	<b>*-0.178</b>	<b>-0.192</b>	<b>0.423</b>
Inf	<b>*-0.154</b>	<b>*-0.151</b>	<b>0.303</b>
Adj	<b>*-0.159</b>	0.137	-0.100
Adv	-0.024	-0.047	0.123
Det	-0.022	<b>-0.243</b>	-0.076
N	<b>*0.177</b>	<b>0.193</b>	<b>-0.433</b>
Prep	0.088	0.049	-0.122
CC	0.065	0.116	-0.086
CS	-0.092	<b>*-0.150</b>	<b>0.459</b>
Pron	0.072	<b>-0.248</b>	0.097
SensWN	<b>-0.285</b>	<b>-0.231</b>	<b>0.236</b>
AmbWN	<b>-0.243</b>	-0.080	<b>*0.154</b>
SensOT	-0.077	-0.093	0.088
AmbOT	<b>-0.208</b>	-0.083	0.099

the unnecessary words (e.g. adjectives, coordinating conjunctions, prepositions) and leaving only the main ideas expressed by verbs.

It is interesting to note that both the average number of senses per word and the percentage of ambiguous words are higher in simplified than in original texts, using both sources (EuroWordNet and Open Thesaurus). One possible explanation (which would have to be explored further) is that the shorter and more commonly used words are more ambiguous than the original words which they substituted in the process of simplification.

#### 4.3 Correlation between Readability Indices and Complexity Features

The Spearman’s rho coefficient of correlation between readability indices and the twelve linguistically motivated complexity features is given in Table 6 (for original texts) and in Table 7 (for simplified texts). Correlations which are significant at a 0.001 level of significance (2-tailed) are presented in bold, while those which are significant at a 0.05 but not at a 0.001 level of significance are presented in bold with an ‘\*’ preceding. Other correlations are not statistically significant.

From the results presented in Table 6 and Table 7 it can be noted that each of the readability indices is significantly correlated with several linguistically motivated complexity features. LC is, for example, positively correlated with occurrences of nouns (*N*) and negatively correlated with occurrences of adjectives (*Adj*) in both corpora. SSR

Table 7: Spearman’s correlation between readability indices and complexity features (Simplified)

Feature	LC	SSR	SCI
V	0.000	-0.059	<b>0.672</b>
Inf	-0.025	-0.074	<b>0.573</b>
Adj	<b>-0.241</b>	0.086	<b>*-0.145</b>
Adv	-0.113	-0.118	<b>0.246</b>
Det	-0.086	<b>-0.438</b>	0.034
N	<b>*0.161</b>	<b>0.375</b>	<b>-0.606</b>
Prep	<b>*0.156</b>	0.088	<b>*-0.153</b>
CC	0.027	0.108	<b>*-0.150</b>
CS	-0.030	<b>*-0.159</b>	<b>*0.595</b>
Pron	0.002	-0.074	<b>-0.186</b>
SensWN	-0.064	-0.070	<b>0.225</b>
AmbWN	-0.110	-0.075	0.115
SensOT	0.053	0.025	0.113
AmbOT	0.110	0.113	0.045

is positively correlated with occurrences of nouns (*N*) and negatively correlated with occurrences of determiners (*Det*) and subordinating conjunctions (*CS*). SCI is, on the other hand, negatively correlated with the number of occurrences of nouns (*N*), and positively correlated with number of occurrences of verbs (*V*), infinitive forms (*Inf*), subordinating conjunctions (*CS*), and average number of senses per word according to Spanish EuroWordNet (*SensWN*).

These results indicate that there is no one readability index which correlates significantly with all of the linguistically motivated complexity features. However, it seems that they complement each other well as each one of them is significantly correlated with a different subset of features. Each of these three readability indices could, therefore, be seen as a measure of a different kind of complexity reduction performed by a text simplification system and thus be used in an automatic evaluation of a text simplification system. That automatic evaluation would, of course, account only for measuring the complexity reduction performed by the system, while a human-oriented evaluation would be needed for assessing the preservation of meaning and grammaticality of the simplified text generated by the system (Drndarevic et al., 2013).

## 5 Conclusions and Future Work

The results presented in this study revealed that there are significant differences between the val-

ues of the three readability indices (LC, SSR, and SCI) applied to the corpus of original news texts and the same applied to manually simplified versions of those texts (aimed at people with cognitive disabilities). Another set of experiments indicated that the two corpora also significantly differ in all but one of the twelve linguistically motivated complexity features.

The study also revealed that the two readability indices which measure lexical complexity of a given text are highly correlated. It also showed that each of the three readability indices (LC, SSR and SCI) significantly correlates with several linguistically motivated complexity features in both corpora. Each of them could thus be used in an automatic evaluation of a text simplification system, each measuring a different kind of complexity reduction performed. Furthermore, it seems that those three readability indices complement each other very well in terms of their correlation with different complexity features. Therefore, it might be possible to find some combination of all three of them which could be used as a single measure in an automatic evaluation of text simplification systems.

The search for this ideal combination will be one of the directions of our future work. We also plan to repeat all these experiments on a different set of texts, this time aimed at a different target population, in order to see whether these readability indices show the same properties for texts simplified in a different manner, i.e. whether they could be used in automatic evaluation of any text simplification system. Furthermore, we wish to apply these indices on texts which were automatically simplified. We would like to explore how well the conclusions drawn based on differences of readability indices between original and automatically simplified texts correlate with human judgments of the level of simplification performed.

## Acknowledgements

This work is partially supported by an Advanced Research Fellowship from Programa Ramón y Cajal (RYC-2009-04291) and by the project SKATER: Scenario Knowledge Acquisition – Knowledge-based Concise Summarization (TIN2012-38584-C06-03), Ministerio de Economía y Competitividad, Secretaria de Estado de Investigación, Desarrollo e Innovación, Spain.

## References

- S. M. Aluísio, L. Specia, T. A. S. Pardo, E. G. Maziero, H. M. Caseli, and R. P. M. Fortes. 2008. A corpus analysis of simple account texts and the proposal of simplification strategies: first steps towards text simplification systems. In *Proceedings of the 26th annual ACM international conference on Design of communication*, SIGDOC '08, pages 15–22, New York, NY, USA. ACM.
- A. Anula. 2007. Tipos de textos, complejidad lingüística y facilitación lectora. In *Actas del Sexto Congreso de Hispanistas de Asia*, pages 45–61.
- M. J. Aranzabe, A. Díaz De Ilaraza, and I. González. 2012. First Approach to Automatic Text Simplification in Basque. In *Proceedings of the first Natural Language Processing for Improving Textual Accessibility Workshop (NLP4ITA)*.
- R. Barzilay and N. Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.
- R. H. M. Brouwer. 1963. Onderzoek naar de leesmoelijkheden van nederlands proza. *Pedagogische studiën*, 40:454–464.
- J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the ACL (EACL'99)*, pages 269–270.
- N. Chomsky. 1986. *Knowledge of language: its nature, origin, and use*. Greenwood Publishing Group, Santa Barbara, California.
- W. Coster and D. Kauchak. 2011. Learning to Simplify Sentences Using Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1–9.
- E. Dale and J. S. Chall. 1948. A formula for predicting readability. *Educational research bulletin*, 27.
- S. Devlin. 1999. *Simplifying natural language text for aphasic readers*. Ph.D. thesis, University of Sunderland, UK.
- W.H. Douma. 1960. De leesbaarheid van landbouwbladen: een onderzoek naar een toepassing van leesbaarheidsformules. *Bulletin*, 17.
- B. Drndarevic, S. Štajner, S. Bott, S. Bautista, and H. Saggion. 2013. Automatic Text Simplification in Spanish: A Comparative Evaluation of Complementing Components. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics. Lecture Notes in Computer Science. Samos, Greece, 24-30 March, 2013.*, pages 488–500.

- W. H. DuBay. 2004. *The Principles of Readability. Impact Information.*
- L. Feng, N. Elhadad, and M. Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL '09*, pages 229–237, Stroudsburg, PA, USA. Association for Computational Linguistics.
- L. Feng. 2009. Automatic readability assessment for people with intellectual disabilities. In *SIGACCESS Access. Comput.*, number 93, pages 84–91. ACM, New York, NY, USA, jan.
- R. Flesch. 1949. *The art of readable writing.* Harper, New York.
- G. Freyhoff, G. Hess, L. Kerr, B. Tronbacke, and K. Van Der Veken, 1998. *Make it Simple, European Guidelines for the Production of Easy-to-Read Information for People with Learning Disability.* ILSMH European Association, Brussels.
- J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom. 1975. Derivation of new readability formulas for navy enlisted personnel. *Research Branch Report 8-75.*
- G. H. McLaughlin. 1969. SMOG grading – a new readability formula. *Journal of Reading*, 22:639–646.
- S. Petersen and M. Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer speech & language*, 23(1):89–106.
- M. B. Ruiter, T. C. M. Rietveld, C. Cucchiari, E. J. Krahmer, and H. Strik. 2010. Human Language Technology and communicative disabilities: Requirements and possibilities for the future. In *Proceedings of the the seventh international conference on Language Resources and Evaluation (LREC).*
- J. Rybing, C. Smith, and A. Silvervarg. 2010. Towards a Rule Based System for Automatic Simplification of Texts. In *The Third Swedish Language Technology Conference.*
- H. Saggion, E. Gómez Martínez, E. Etayo, A. Anula, and L. Bourg. 2011. Text Simplification in Simplext: Making Text More Accessible. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural.*
- S. E. Schwarm and M. Ostendorf. 2005. Reading Level Assessment Using Support Vector Machines and Statistical Language Models. In *Proceedings of the 43rd annual meeting of the Association of Computational Linguistics (ACL)*, pages 523–530.
- E. A. Smith and R. J. Senter. 1967. Automated Readability Index. Technical report, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio.
- S. Spaulding. 1956. A Spanish Readability Formula. *Modern Language Journal.*
- P. van Oosten, D. Tanghe, and V. Hoste. 2010. Towards an Improved Methodology for Automated Readability Prediction. In *Proceedings of the seventh international conference on language resources and evaluation (LREC10).* Valletta, Malta: European Language Resources Association (ELRA).
- P. Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks.* Kluwer Academic Publishers.
- S. Štajner, R. Evans, C. Orasan, and R. Mitkov. 2012. What Can Readability Measures Really Tell Us About Text Complexity? In *Proceedings of the LREC'12 Workshop: Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, Istanbul, Turkey.
- K. Woodsend and M. Lapata. 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP).*
- S. Wubben, A. van den Bosch, and E. Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, ACL '12*, pages 1015–1024, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Z. Zhu, D. Bernard, and I. Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.

# Weasels, Hedges and Peacocks: Discourse-level Uncertainty in Wikipedia Articles

Veronika Vincze<sup>1,2</sup>

<sup>1</sup>Hungarian Academy of Sciences, Research Group on Artificial Intelligence

<sup>2</sup>Department of Informatics, University of Szeged

vinczev@inf.u-szeged.hu

## Abstract

Uncertainty is an important linguistic phenomenon that is relevant in many areas of language processing. While earlier research mostly concentrated on the semantic aspects of uncertainty, here we focus on discourse- and pragmatics-related aspects of uncertainty. We present a classification of such linguistic phenomena and introduce a corpus of Wikipedia articles in which the presented types of discourse-level uncertainty – weasel, hedge and peacock – have been manually annotated. We also discuss some experimental results on discourse-level uncertainty detection.

## 1 Introduction

In many areas of natural language processing, it is essential to distinguish between factual and non-factual information. Thus, depending on the precise task, negated or uncertain propositions should be treated separately by e.g. information extraction systems or they should be neglected. For instance, in medicine, if it is uncertain whether the patient suffers from an illness, the doctor should undertake further examinations to determine the final diagnosis. In another case, only those pieces of news are relevant in news media that are true and come from a reliable source. Uncertain information or unreliable sources should not be part of the news. In order to be able to find uncertain propositions in a huge amount of texts, a reliable uncertainty detector is needed, which can be only developed if annotated resources are at hand.

Previous studies on uncertainty detection concentrated mostly on the semantic dimensions. Indeed, in many cases it is the lexical content (meaning) of the uncertainty marker (cue) that is responsible for uncertainty, i.e. it can be identified

in texts with the help of semantic tools. However, there are other types of uncertainty which cannot be described by just concentrating on semantics. For instance, *many* may denote quite different approximations: in the sentence *Many of the students did not read the book*, *many* may signal about 60-70% of the students (or at least more than 50%), while in *This airline loses many suitcases*, *many* may be only 20% but this number is still high enough for passengers to call it *many*. Here, the context and world knowledge determine how the quantifier *many* should be interpreted.

Here, we will focus on pragmatics- and discourse-related aspects of uncertainty. We will examine the concepts of source, fuzziness and subjectivity and their connection with uncertainty. As a first contribution, we will present a language-independent classification of such linguistic phenomena. As another contribution, we will introduce a corpus of Wikipedia articles in which linguistic cues of the presented types of discourse-level uncertainty have been manually annotated, hence empirical data on the frequency of such phenomena can also be provided. We will report the results of our experiments and we will also compare them with those of previous studies.

## 2 Discourse-level Uncertainty

Different concepts and terms that are related to uncertainty phenomena are employed. Modality is usually associated with uncertainty (Palmer, 1986), but the terms factuality (Saurí and Pustejovsky, 2012), veridicality (de Marneffe et al., 2012), evidentiality (Aikhenvald, 2004) and commitment (Diab et al., 2009) are also used. They all represent related but slightly different linguistic phenomena, which lie mostly in the category of semantic uncertainty. Propositions can be uncertain at the semantic level, that is, their truth value cannot be determined just given the speaker's mental state. Szarvas et al. (2012) offer a classi-

fication of semantic uncertainty phenomena.

Here, we use the term uncertainty similar to Szarvas et al. (2012), who aimed at giving a unified framework for the above-mentioned phenomena: “uncertain propositions are those [...] whose truth value or reliability cannot be determined due to lack of information”. They contrast semantic uncertainty with discourse-level uncertainty: if the scheme “*cue x* but it is certain that not *x*” is invalid (where *x* denotes a proposition, and *cue* denotes an uncertainty cue), that is, an uncertain proposition and its negated version cannot be coordinated, it is an instance of semantic uncertainty (e.g. *##It may be raining in New York but it is certain that it is not raining in New York*).

Besides semantic uncertainty, uncertainty can be found at the level of discourse as well. Here, the missing or intentionally omitted information is not related to the propositional content of the utterance but to other factors. In contrast to semantic uncertainty (Szarvas et al., 2012), the truth value of such propositions can be determined, but uncertainty arises if the proposition is analyzed in detail. For instance, the sentence *Some people are running* evokes questions like *Who exactly are those people that are running?* Here, the answer usually depends on the context, the speaker and the discourse and it cannot be determined out of context, thus henceforth such phenomena will be labeled discourse-level uncertainty.

We will carefully analyze discourse-level uncertainty phenomena below which are named after their most typical linguistic markers, i.e. cues. Although for the sake of simplicity we only provide English examples here, our categorization is based on pragmatic and cognitive considerations, and we will implicitly assume that our categories are language-independent. We will focus on Wikipedia articles, which – as indicated by previous studies (Ganter and Strube, 2009; Farkas et al., 2010) – seem to contain a certain amount of uncertainty phenomena like this. We will concentrate on three key aspects of discourse-level uncertainty, namely, sources, fuzziness and subjectivity.

## 2.1 Weasels

The notion of source is important for deciding the reliability of information conveyed (Saurí and Pustejovsky, 2012; Wiebe et al., 2005; Nawaz et al., 2010). It is not a matter of indifference to whom the information / opinion belongs to, espe-

cially in news media: people are more likely to believe a statement if it is communicated by a reliable source as opposed to a piece of sourceless information. In the public mind, experts, scientists, ministers, etc. are viewed as credible sources (cf. Bell (1991)) while unnamed or unidentifiable sources are considered less reliable. If some pieces of information are backed by a credible source, they are more likely to be treated as trustworthy, however, sourceless information is given less credence.

Events with no obvious sources are called *weasels* in Wikipedia<sup>1</sup> (Ganter and Strube, 2009): their source is missing or is specified only vaguely or too generally, hence, it cannot be exactly determined who the holder of the opinion is (undetermined source) as it is either not expressed or expressed by an indefinite noun phrase. Weasel sentences usually invoke questions like *Who said that?* and *Who thinks that?* The following sentence illustrates this:

**Some** have claimed that Bush would have actually increased his lead if state wide recounts had taken place.

The ultimate source of the proposition expressed in the embedded sentence is not known since it is denoted by the pronoun **some**. Thus, it is not known who provided the opinion and therefore it is uncertain whether this is an important (reliable) piece of information (e.g. the opinion of experts) or whether it should be ignored.

Passive constructions which do not express the agent comprise a special type of weasels:

It has been suggested [**by whom?**] that he should have involved Clinton much more heavily in his campaign.

The sentence does not reveal who has suggested the involvement of Clinton in the campaign. Hence, the source of the information is unclear and the source is missing from the sentence.

The basic idea behind weasel phenomena is the lack of a reference: it is not known who the source of the opinion is. This view is supported by the fact that a weasel candidate ceases to be uncertain if it is enhanced by citations:

Most authors now prefer to place it within the genus *Pezoporus*, e.g. Leeton et al. (1998).

<sup>1</sup>[http://en.wikipedia.org/wiki/Wikipedia:Manual\\_of\\_Style/Words\\_to\\_watch](http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Words_to_watch)

The phrase *most authors* would indicate a weasel (it is not clear whose opinion is this) but the citation at the end of the sentence clearly identifies the source.

In this paper, we extend the original notion of *weasel* and we argue that propositions that have an underspecified argument that would be relevant or is not common knowledge in the situation can be also viewed as weasels. Thus, a proposition is considered to be an instance of weasel if any of its relevant arguments is underspecified, i.e. it evokes questions like *Who/what exactly? Which?* Here, we give an example:

While the Skyraider is not as iconic as **some other** aircraft, it has been featured in some Vietnam-era films such as *The Green Berets* (1968) and *Flight of the Intruder* (1991).

The sentence does not determine what kind of aircraft is considered iconic, so it is a vague or underspecified statement: we only know that there are “iconic aircraft”, but no more details are specified. Again, the weasel type of uncertainty is expressed here by the adjectives *some* and *other*. Note that there is another occurrence of the word *some* in the sentence, but it does not denote any uncertainty in this case since the relevant Vietnam-era films are then listed.

## 2.2 Hedges

Another type of discourse-level uncertainty that will be discussed later on is called a hedge. Although a lot of studies used the term *hedge*, it may denote different linguistic phenomena for different authors. For instance, *hedge* means mostly *speculation* in the biomedical domain (see e.g. Medlock and Briscoe (2007), Vincze et al. (2008), and Farkas et al. (2010)). When contrasting epistemic modality and hedging, Rizomilioti (2006) categorizes approximators, passive voice and attribution to unnamed sources, among others, as instances of hedging and Hyland (1996) also cites them among common hedging devices.

Here, we understand *hedge* in the sense introduced by Lakoff (1973). For him, hedges are “words whose job is to make things fuzzier or less fuzzy”, that is, the exact meaning of some qualities or quantities is blurred by them. Intensifiers (*very*, *much*), deintensifiers (*a bit*, *less*) and circumscribers (*approximately*) also belong to this

group. Their effect is to add uncertainty to some elements in the proposition: they shift the value of some quality / quantity and the truth value of the proposition can only be decided if it is known what the reference point in the discourse is as the following example shows:

Specialized services will **very often** provide a **much** more reliable service based on trusted publications.

In this sentence, there are several hedge cues. First, there is *often*, which informs us that it is not always the case that specialized services provide much more reliable service. It is modified by the intensifier *very*, which indicates that it is almost always the case (but still not always). Next, their service is *much* more reliable than any other service (at least those relevant in the context), that is, it is very reliable.

However, it should be noted that there is no absolute way to determine the truth value of this proposition without agreeing on what is meant by e.g. *often*: for now, let us say that *often* means at least seven out of ten times (but not ten times out of ten) and then *very often* may denote eight or nine times out of ten. It depends on the context, the speakers and the event described in the sentence to determine the reference point according to which the quantity or quality of events or entities can be evaluated. In the above example, the reference point may be 70%, and intensifiers denote that the quality or frequency of the event / entity is above the reference point, in this case, above 70%. Deintensifiers, however, assert that the quality or frequency is below the reference point.

Circumscribers – as their name states – circumscribe the exact amount or quality of the event or entity, which can be above or below the reference point. To represent this visually, they denote a set around the reference point in which the exact amount or quality is situated (see Figure 1 below). Here are some linguistic examples:

This may explain why it has a lower than average estimated albedo of **~0.03**.

The duration of attacks averages **3-7** days.

It is interesting to note that in such cases not only cue words but also cue characters are responsible for uncertainty: the tilde and hyphen in these specific cases. Moreover, there are cue words that

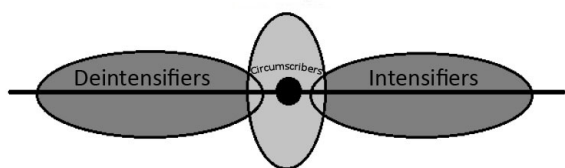


Figure 1: Types of hedges.

function as circumscibers as well like *approximately* and another use of *some*:

Amsterdam Zuidoost has **approximately** 86,000 inhabitants and consists of **some** 38,000 houses.

Figure 1 shows the relationship of hedge types relative to the reference point. Thus, each type of hedge denotes a set in which the exact amount, quality or frequency of the relevant event or entity is situated but its exact place remains unclear.

Hedging is also one of the politeness strategies mentioned by Brown and Levinson (1987): they may function as mitigators in order to minimize disagreement, and to acknowledge that the speaker is imposing a task on the hearer. In the request *Could you please sort of correct this very short text for me?* the phrase *sort of* is a hedge, and the “very short” text may in fact be rather long. Here, hedges have pragmatic functions and they do not refer to uncertainty.

### 2.3 Peacocks

Subjectivity by its very nature contains aspects of uncertainty. People’s opinions may differ from each other concerning specific things or events: they do not necessarily agree on what is good, neutral or bad. Thus, we cannot unequivocally determine what is good or what is bad.

Words that express unprovable qualifications or exaggerations are called *peacock* by Wikipedia editors.<sup>2</sup> Their meaning often inherently contain positive or negative subjective judgments, that is, they are polar expressions. Peacock terms include *brilliant*, *excellent* and *best-known*. Although their usage may be acceptable in other contexts, the objective style of Wikipedia editing requires that peacocks should be avoided.

<sup>2</sup>[http://en.wikipedia.org/wiki/Wikipedia:Manual\\_of\\_Style/Words\\_to\\_watch](http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Words_to_watch)

Although they are not called peacocks by Wikipedia editors, we classify other subjective elements as peacocks as well. For instance, editorial remarks that refer to the subjective opinion of the author of the article (like *ironically* and *unfortunately*) or contentious labels (*controversial* and *legendary*) may all express subjectivity in certain contexts, hence we treat them here as peacock terms. The uncertainty in their meaning again lies in the fact that it cannot be objectively judged what can be called *excellent* for instance – it can be only deduced from discourse or contextual information and it may differ from speaker to speaker.

Here is a sentence with some peacock terms:

Through the **ardent** efforts of Rozsnyai, the Philharmonia Hungarica quickly matured into one of Europe’s **most distinguished** orchestras.

The words *ardent* and *most distinguished* are clearly positive in polarity, and again it cannot be objectively decided what level of enthusiasm is called ardent or which orchestras belong to the most distinguished ones.

All peacock terms are similar to hedges to some extent. They can be called scalar uncertainties since in both cases, a scale is involved in the interpretation of the uncertain term. In the case of peacock, there is a scale of polarity on which phrases can be judged as positive or negative whereas in the case of hedges, there is a scale on which there is a reference point, on the basis of which the uncertain part of the utterance is placed. Although they are similar, we suggest that peacocks and hedges be differentiated in our classification because peacocks are related to subjectivity while hedges are more neutral, hence they can be relevant for different NLP applications (e.g. in opinion mining, which seeks to collect subjective opinions on different topics, peacocks may prove more useful than hedges). Still, hedges shift the value of the quantity / quality mentioned in the text while peacocks denote a specific point on the scale, without modifying it, which again suggests that they should not be lumped in the same class.

### 3 Related Work

These days, uncertainty and modality detection is a widely studied area in natural language processing, which manifests itself in a number of corpora annotated for uncertainty in domains like biology

(Medlock and Briscoe, 2007; Kim et al., 2008; Vincze et al., 2008; Nawaz et al., 2010), medicine (Uzuner et al., 2009), news media (Wilson, 2008; Saurí and Pustejovsky, 2009; Rubin, 2010), and encyclopedia texts (Farkas et al., 2010). Although some authors have called attention to the fact that the progressive nature of discourse and dimensions of time should be also taken into account (de Marneffe et al., 2012; Saurí and Pustejovsky, 2012), as can be judged on the basis of available guidelines, most of these corpora make use of semantic uncertainty, with some exceptions that take into account pragmatic or discourse-level information as well (see below).

The concept of source has played a significant role in the literature. FactBank (Saurí and Pustejovsky, 2009) explicitly annotates the factuality of events according to their sources' perspective and Wiebe et al. (2005) also emphasize the role of sources annotated in the MPQA corpus for opinion mining. The notion of perspective – both in Nawaz et al. (2010) and in Morante and Daelemans (2011) – is similar to the one of sources applied in FactBank and MPQA. In Wikipedia, the lack of identifiable sources is explicitly discouraged by editors. They call such phenomena *weasels* (see also Ganter and Strube (2009)) and weasel detection was one of the subtasks of the CoNLL-2010 shared task (Farkas et al., 2010).

The lack of source characteristics to weasels can be paired with a certain strategy that Hyland (1996) calls impersonal constructions. It is a type of writer-oriented hedges<sup>3</sup> in his system. It is interesting to note that in his system, the opposite of this strategy can also be found, which could be called *anti-weasel*: the writer emphasizes his responsibility by using first person pronouns. However, this latter strategy does not represent any form of uncertainty in our view.

Fuzziness is another dimension of uncertainty. Lakoff (1973) gave an account of some lexical items – which he calls hedges – that “make things fuzzier”, that is, words such as *approximately*, *kind of*, *at least* etc. Due to the presence of such words, the quality or quantity under investigation is shifted on a scale. If modified by the adverb *very* for instance, it moves towards one end of the scale on which this quality/quantity is determined. The phenomenon of hedging in scientific articles

---

<sup>3</sup>However, in our classification, it should be called a weasel.

is analyzed and categorized according to the functions it can fulfill in Hyland (1996).

Subjectivity is also related to uncertainty. There is a great diversity among individual views and opinions: a feature of a product may be appreciated by some customers but it might be considered intolerable for others. Thus, what should be considered positive or negative seems subjective. Many approaches to subjectivity or sentiment analysis rely on lexicons and databases of subjective terms. For instance, the database SentiWordNet (Baccianella et al., 2010) contains a subset of the synsets of the Princeton Wordnet with positivity, negativity and neutrality scores assigned to each concept, depending on the use of its sentiment orientation, thus it is a lexicon where subjective terms are listed and ranked. Wilson (2008) defines subjectivity clues as words and phrases that express private states, that is, individual opinions. She distinguishes lexical cues and syntactic cues that are responsible for subjectivity. She lists several modifiers among her syntactic clues of subjectivity like *quite* and *really*. However, in contrast with other subjective elements, we do not regard them as peacock cues since – as Wilson (2008) herself states – they “work to intensify”, so in our system they are classified as hedge cues. On the other hand, some instances of biased language can also be classified as peacocks in our system (Recasens et al., 2013).

Human communication and discourse is incremental in nature (Cristea and Webber, 1997). Information may be added at a later point of the discourse that clarifies a previously missing piece of information. Applying this to discourse-level uncertainty, it may be the case that an apparent weasel phrase is elaborated on later in the discourse, or the exact value of an apparent hedge expression is later provided. In such cases, the phrases should not be marked as uncertain, which indicates the essential role of co-text – i.e. surrounding words in the text (Brown and Yule, 1983) – in detecting discourse-level uncertainty.

## 4 The Annotated Corpus

In order to test the practical applicability of the new classification of discourse-level uncertainty phenomena, and to investigate the frequency of each uncertainty type, we also created an annotated corpus. We selected WikiWeasel, the Wikipedia subset of the CoNLL-2010 Shared



Task corpora (Farkas et al., 2010) for annotation. By doing this, our results could be contrasted with those of the original annotation carried out specifically for the shared task. Moreover, as the corpus has recently been annotated for semantic uncertainty (Szarvas et al., 2012), interesting comparisons can also be made between semantic and discourse-level uncertainty. The annotated corpus is available free of charge for research purposes at [www.inf.u-szeged.hu/rgai/uncertainty](http://www.inf.u-szeged.hu/rgai/uncertainty).

#### 4.1 Statistical Data on the Corpus

The dataset consists of 4,530 Wikipedia articles and 20,756 sentences. Texts were manually annotated by two linguists for linguistic cues denoting all types of discourse-level uncertainty, i.e. weasel, peacock and hedge. 200 articles were annotated by both linguists and the inter-annotator agreement rate for the categories weasel, peacock and hedge were 0.4837, 0.4512 and 0.4606, respectively (in terms of  $\kappa$ -measure), which reflects that identifying discourse-level phenomena is not straightforward, however, it can be reasonably well solved considering the subjective nature of the task. During the annotation, special emphasis was laid on the discourse structure of the text. For instance, weasel cue candidates do not denote uncertainty when the sentence is enhanced with citations. Also, a weasel-like element may be elaborated on in the next sentence, thus it is not to be marked as weasel as in:

Some ship names are references to other games created by Jordan Weisman. The “Black Swan” is a reference to a character from *Crimson Skies*, and also possibly to the ship *Black Pearl* from *Pirates of the Caribbean*.

In order to attain the gold standard for the commonly annotated parts, the two annotators discussed problematic cases and reached a consensus for each case. The final version of the corpus contains these disambiguated cases.

The dataset contains 10,794 discourse-level uncertainty cues<sup>4</sup>, which occur in 7,336 uncertain

<sup>4</sup>We should mention that our corpus contained 680 passive constructions, which were annotated as weasels. As we focus now on lexical cues of discourse-level uncertainty, and they belong to syntactic cues, the investigation of such cases will be subject to further studies.

sentences. A sentence was considered to be uncertain if it contained at least one uncertainty cue. But, as the results show, many sentences include more than one uncertainty cue. Statistical data on the uncertainty cues found in the WikiWeasel corpus are listed in Table 1, together with available data on semantic uncertainty types, taken from Szarvas et al. (2012).

Uncertainty cue	#	%	Diff. cues
Hedge	4,743	35.24	260
Weasel	4,138	30.75	99
Peacock	1,913	14.21	540
Discourse-level total	10,794	80.2	899
Epistemic	1,171	8.7	114
Doxastic	909	6.75	36
Conditional	491	3.65	15
Investigation	94	0.7	12
Semantic level total	2,665	19.8	166
Total	13,459	100	1065

Table 1: Uncertainty cues in WikiWeasel.

As can be seen, most of the uncertainty cues found in the corpus belong to the discourse-level uncertainty class, the ratio of semantic to discourse-level uncertainty cues being 1:4. Among the types of discourse-level uncertainty, hedges are the most frequent, followed by weasels and peacocks. All this suggests that discourse-level uncertainty is very typical of Wikipedia articles, about 35% of the sentences being uncertain at the discourse level. As regards the specific classes, 3,807 (18.3%), 3,497 (16.8%) and 1,359 (6.5%) sentences contain at least one hedge, weasel or peacock cue, respectively.

#### 4.2 Cue Distribution in the Corpus

On the number of different cues, Table 1 tells us that the set of linguistic cues expressing weasels are the most limited, with almost 100 cues. In contrast, peacock cues vary the most with 540 cues. This suggests that weasels have the most restricted vocabulary in contrast to peacocks, and hedges being in the middle. This also means that the average frequency of a weasel cue is much higher than that of a peacock cue: the average frequency of occurrence of weasel, hedge and peacock cues is 41.8, 18.24 and 3.54, respectively.

We did a more detailed analysis on the lexical distribution of the cues as well. The ten most frequent cues for each type are listed in Table 2. These are responsible for about 86%, 45% and 42% of the occurrences of weasel, hedge and peacock cues, respectively. Thus, a limited vocabu-

Weasel	#	%	Hedge	#	%	Peacock	#	%
some	887	25.64	often	539	11.36	most	318	16.62
many	631	18.24	usually	263	5.55	popular	112	5.85
other	539	15.58	many	217	4.58	famous	81	4.23
several	204	5.90	generally	210	4.43	well-known	50	2.61
most	202	5.84	very	206	4.34	notable	50	2.61
various	177	5.12	most	179	3.77	notably	45	2.35
others	175	5.06	almost	152	3.20	important	40	2.09
certain	82	2.37	several	140	2.95	best	38	1.99
number	43	1.24	common	127	2.68	traditionally	38	1.99
critics	37	1.07	much	119	2.51	controversial	37	1.93

Table 2: The most frequent discourse-level uncertainty cues in the WikiWeasel corpus.

lary can account for over 85% of weasels.

However, some terms can belong to more than one uncertainty type. For example, *most* occurs in all the three types (weasel: *Most agree that this puts her at about 12 years of age*, hedge: *He spent most of his time working on questions of theology* and peacock: *Kathu is the district which covers the most touristical beach of Phuket*), but *some*, *many* and *several* can all be instances of weasels and hedges. This is due to the linguistic variability of these items: e.g. *some* may refer to “an indefinite quantity” or “something unspecified”.

As can be seen, there are some overlapping cues among the types. This is especially so in the case of hedges and weasels: 25 cues can denote hedges or weasels as well, thus 25% of the weasel cues are ambiguous. These cues were also responsible for most of the differences between the two annotations, which indicates that their identification requires special attention both for human annotators and NLP tools: it is mostly the neighbouring words that can determine whether it is a weasel or hedge. For instance, if *some* occurs before a verb and constitutes a noun phrase on its own, then it is almost certainly a weasel cue (*Some think that...*) but if it occurs before a noun denoting time, it is probably a hedge (*some minutes ago*).

## 5 Experiments

We carried out some baseline experiments on the corpus. We divided the corpus into training (80%) and test (20%) sets and applied a simple dictionary-based approach which classified each cue candidate as uncertain if it was tagged as uncertain in at least 50% of its occurrences in the training dataset. For ambiguous cues, the most frequent label was chosen (e.g. *most* was used as a peacock cue). Similar to the CoNLL-2010 shared task, we evaluated our results at the cue level as well as at the sentence level.

	Cue level			Sentence level		
	P	R	F	P	R	F
Weasel	0.7088	0.6724	0.6901	0.7443	0.7183	0.7311
Hedge	0.8780	0.6616	0.7546	0.9185	0.7193	0.8068
Peacock	0.4222	0.4730	0.4462	0.4034	0.5341	0.4597
Micro F	0.7196	0.6348	0.6745	0.7458	0.6924	0.7181

Table 3: Baseline results in terms of precision / recall / F-score.

Table 3 shows that the peacock class is the most difficult to detect, which may be due to the fact that this class has the most diverse cues and thus applying a dictionary-based method leads to a lower recall. Still, the lower precision was due to the higher level of ambiguity concerning the most typical peacock cues (like *most*). As for hedges, a simple lexical approach can result in a good precision score, which suggests that hedge cues are less ambiguous than weasel or peacock cues. It is also seen that sentence-level results are significantly higher than cue-level results (ANOVA,  $p = 0.0026$ ). Uncertain sentences typically contain more than one cue and in the former scenario, it is sufficient to recognize only one cue in the sentence to regard the sentence as uncertain and false negatives do not affect the performance significantly.

If we compare the data with the CoNLL-2010 version of the corpus, it is seen that the new annotation scheme leads to many more cues (6,725 cue phrases in 4,718 uncertain sentences in the original version vs. 10,794 cues in 7,336 sentences in the version described here) and – although the datasets are not directly comparable – it gives a much better performance: the best system achieved an F-score of 60.2 on weasel detection at the sentence level and 36.5 at the cue level and no classes of cues were distinguished there (Farkas et al., 2010). This difference may be attributed to several factors. First, not all hedge phenomena (used in the sense introduced here) were systematically annotated in the CoNLL-2010 corpus.

Second, complex syntactic structures that contained several types of uncertainty were annotated as one complex cue (e.g. the phrase *it has been widely suggested*, which contains epistemic uncertainty (*suggested*), weasel (passive sentence with no agent) and hedge (*widely*) as well). Third, the CoNLL-2010 version did not distinguish subtypes of cues, i.e. semantic uncertainty and weasels were annotated in the same way. It was probably because of this lack of distinction that participants of the shared task got considerably lower results for Wikipedia articles than for biological papers, which contained fewer weasel cues (Farkas et al., 2010). However, the new annotation makes it possible to select those types of uncertainty that are relevant for a given application, see Section 6.

## 6 Discourse-level Uncertainty and NLP

Detecting weasels is of utmost importance in every information extraction application where it should be known who the author/source is. Thus, information extraction applied for the news media may certainly profit from finding weasels, i.e. missing or undeterminable sources. Pieces of information without an identifiable (and reliable) source require special treatment: they will be excluded from the news or they will be communicated to the public in a special form, using phrases such as *according to unnamed sources* etc.

In sentiment analysis and opinion mining, the identification of subjective terms is essential. These terms are often ambiguous hence a subjectivity word sense disambiguation is needed (Wiebe, 2012). In our corpus, peacock terms and intensifiers – a subtype of hedges – are manually annotated, thus it can be used in the development and evaluation of tools that seek to disambiguate elements of a subjectivity lexicon in running texts.

Information retrieval may also be enhanced by detecting discourse-level uncertainty. In order to find relevant documents for queries that contain numbers, more specifically, to improve recall in such cases, it is important to handle numeric hedges. For instance, if someone looks for websites describing games appropriate for ten year old children, he also may be interested in games that are for children over eight. Thus, the search engine should be prepared for recognizing that the number specified in the query (“ten”) is part of other numeric sets (e.g. “over eight”) and in this way, more relevant hits can be retrieved.

The linguistic processing of patents especially requires that hedges should be recognized. There is a tendency to generalize over the scope of the patent (i.e. hedges are used) in order to prevent further abuse (Osenga, 2006). Thus, the scope of the patents can be expanded or other use cases can later be included in the patent. Hence, any NLP system that aims at patent processing must target hedge detection as well.

Document classification may also profit from detecting discourse-level uncertainty since different genres of texts involve different types of uncertainty. For instance, papers in the humanities contain significantly more hedges than papers in sciences (Rizomilioti, 2006). Thus, the frequency of hedges may be indicative of the domain of the text as well, which again may be exploited in document classification.

## 7 Conclusions

In this paper, we presented a classification of discourse-level uncertainty phenomena, and focused on the concepts of source, fuzziness and subjectivity. We also introduced a corpus of Wikipedia articles in which linguistic cues for each type of discourse-level uncertainty – weasel, peacock and hedge – were manually annotated. We carried out some baseline experiments on discourse-level uncertainty detection, which may prove useful in information extraction and retrieval, sentiment analysis and opinion mining.

In the future, we intend to develop a machine-learning based uncertainty detector. We would also like to investigate the distribution of weasels, hedges and peacocks in other types of texts (e.g. news media or scientific papers) and in other languages as our three categories are language-independent. Moreover, to learn how domain-dependent the model is, we plan to do some domain adaptation experiments as well.

## Acknowledgments

This work was supported in part by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013).

## References

Alexandra Y. Aikhenvald. 2004. *Evidentiality*. Oxford University Press, Oxford.

- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of LREC'10*, Valletta, Malta, May. ELRA.
- Allan Bell. 1991. *The language of the News Media*. Blackwell, Oxford.
- Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some universals in language usage*. CUP, Cambridge, UK.
- Gillian Brown and George Yule. 1983. *Discourse Analysis*. CUP, Cambridge, UK.
- Dan Cristea and Bonnie Webber. 1997. Expectations in incremental discourse processing. In *Proceedings of ACL97/EACL97*, pages 88–95. Morgan Kaufmann.
- Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2012. Did it happen? The pragmatic complexity of veridicality assessment. *Computational Linguistics*, 38:301–333.
- Mona T. Diab, Lori S. Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran, and Weiwei Guo. 2009. Committed belief annotation and tagging. In *Proceedings of LAW 2009*, pages 68–73. The Association for Computer Linguistics.
- Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of the CoNLL-2010 Shared Task*, pages 1–12, Uppsala, Sweden, July. ACL.
- Viola Ganter and Michael Strube. 2009. Finding Hedges by Chasing Weasels: Hedge Detection Using Wikipedia Tags and Shallow Linguistic Features. In *Proceedings of ACL-IJCNLP 2009*, pages 173–176, Suntec, Singapore, August. ACL.
- Ken Hyland. 1996. Writing without conviction? Hedging in scientific research articles. *Applied Linguistics*, 17(4):433–454.
- Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(Suppl 10).
- George Lakoff. 1973. Hedges: A study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2(4):458–508.
- Ben Medlock and Ted Briscoe. 2007. Weakly Supervised Learning for Hedge Classification in Scientific Literature. In *Proceedings of the ACL*, pages 992–999, Prague, Czech Republic, June.
- Roser Morante and Walter Daelemans. 2011. Annotating Modality and Negation for a Machine Reading Evaluation. In *Proceedings of CLEF 2011*.
- Raheel Nawaz, Paul Thompson, and Sophia Ananiadou. 2010. Evaluating a meta-knowledge annotation scheme for bio-events. In *Proceedings of NESP 2010*, pages 69–77, Uppsala, Sweden, July. University of Antwerp.
- Kristen Osenga. 2006. Linguistics and Patent Claim Construction. *Rutgers Law Journal*, 38(61):61–108.
- Frank Robert Palmer. 1986. *Mood and Modality*. CUP, Cambridge.
- Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of ACL-2013*, pages 1650–1659, Sofia, Bulgaria, August. ACL.
- Vassiliki Rizomilioti. 2006. Exploring epistemic modality in academic discourse using corpora. In *Information Technology in Languages for Specific Purposes*, pages 53–71. Springer US.
- Victoria L. Rubin. 2010. Epistemic modality: From uncertainty to certainty in the context of information seeking as interactions with texts. *Information Processing & Management*, 46(5):533–540.
- Roser Saurí and James Pustejovsky. 2009. FactBank: a corpus annotated with event factuality. *Language Resources and Evaluation*, 43:227–268.
- Roser Saurí and James Pustejovsky. 2012. Are you sure that this happened? Assessing the factuality degree of events in text. *Computational Linguistics*, 38:261–299.
- György Szarvas, Veronika Vincze, Richárd Farkas, György Móra, and Iryna Gurevych. 2012. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38:335–367.
- Özlem Uzuner, Xiaoran Zhang, and Tawanda Sibanda. 2009. Machine Learning and Rule-based Approaches to Assertion Classification. *Journal of the American Medical Informatics Association*, 16(1):109–115, January.
- 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(Suppl 11):S9.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):164–210.
- Janyce Wiebe. 2012. Subjectivity word sense disambiguation. In *Proceedings of WASSA 2012*, page 2, Jeju, Korea, July. ACL.
- Theresa Ann Wilson. 2008. *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States*. Ph.D. thesis, University of Pittsburgh, Pittsburgh.

# Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia

**Fahd Alotaibi**

School of Computer Science  
University of Birmingham, UK  
f.s.a081@cs.bham.ac.uk

**Mark Lee**

School of Computer Science  
University of Birmingham, UK  
m.g.lee@cs.bham.ac.uk

## Abstract

This paper presents a methodology to exploit the potential of Arabic Wikipedia to assist in the automatic development of a large Fine-grained Named Entity (NE) corpus and gazetteer. The corner stone of this approach is efficient classification of Wikipedia articles to target NE classes. The resources developed were thoroughly evaluated to ensure reliability and a high quality. Results show the developed gazetteer boosts the performance of the NE classifier on a news-wire domain by at least 2 points F-measure. Moreover, by combining a learning NE classifier with the developed corpus the score achieved is a high F-measure of 85.18%. The developed resources overcome the limitations of traditional Arabic NE tasks by more fine-grained analysis and providing a beneficial route for further studies.

## 1 Introduction

Previous efforts that have been made to develop an Arabic NER either focused on traditional NE classes (Benajiba et al., 2010) or sought to expand only one class at a time (Shaanan and Raza, 2007). Applications such as Question Answering (QA) receive more benefits when a fine-grained NER is developed. This is true when we consider that, the majority of factoid questions are about named entities (Noguera et al., 2005). Having a finer NER, results in the possibility of extracting more semantic knowledge from the context. For example, if we consider the following sentence:

شركة والت ديزني هي أكبر شركات وسائل الإعلام والترفيه في العالم /šrkħ wAlt dyzny hy Okbr šrkAt wsAÿl AllçlAm wAltrfyh fy AlçAlm/ 'Walt Disney is the largest media company in the entertainment world'<sup>1</sup>

We would have more semantic information if we could tag (الت ديزني /wAlt dyzny/ 'Walt Disney')

<sup>1</sup>Throughout this paper, Arabic words are represented in three variants: (Arabic word /HSB transliteration scheme (Habash et al., 2007) / "English translation")

as [ORG-ENTERTAINMENT] rather than just [ORG]. This deeper semantics is very helpful when answering factoid question like "What is the largest entertainment company?"

Supervised machine learning technologies have been successfully adopted for several natural language tasks, including NER. These technologies require a reasonable portion of data to be accessible in the training phase, containing a number of positive and negative examples to learn from and to circumvent the problem of data sparseness. Traditional methods for compiling such data involve recruiting individuals to annotate a certain corpus manually. This is tedious work, as well as costly and time consuming. Moreover, manually annotating a large portion of a relatively open domain corpus beyond a news-wire and across various genres is not easy for an individual to achieve.

Therefore, when developing a reasonable fine-grained NE corpus two questions should be answered. First, *what proper fine-grained semantic classes should be established?* Second, *how to develop a reasonable sized fine-grained NE corpus at minimum cost?* This work answers those questions.

To these ends the methodology we devised was designed to utilise the availability and growth of Arabic Wikipedia to develop a large and extendable fine-grained named entity corpus and gazetteer with minimum human intervention. The contributions of this paper are:

1. It introduces a two-level tagset for Wikipedia NEs;
2. It develops a large fine-grained automatic NE corpus using minimum human intervention;
3. It develops a large fine-grained gazetteer; and
4. It thoroughly evaluates the resulting corpus and gazetteer.

## 2 Arabic Wikipedia and Named Entity

Wikipedia is an extensive collaborative project on the web in which articles are published and reviewed by volunteers from around the world. Wikipedia includes

271 different languages, with the Arabic version ranked 27th with more than 210,000 articles. The annual increase in the number of articles is 30% (Wikipedia, 2013). The actual relationship between the Named Entity and Wikipedia is that a large percentage of Wikipedia articles are about named entities (Alotaibi and Lee, 2012). This provided the motivation to utilise Wikipedia’s underlying structure to produce the target resources.

To this end, it is beneficial to provide an overview of the critical aspects of the Wikipedia structure:

- **Articles:** These can be one of the following:

1. **Normal article:** Each article has a unique title and contains authentic content; i.e. textual data, images, tables, items and links, related to the concept represented in the title. These are in the majority.
2. **Redirected article:** These contain a specific tag to redirect the enquirer to a normal article. For example: for the redirected article titled (بريطانيا العظمى /bryTAnyA AlçĎmý/ ‘Great Britain’), there is a redirected tag to (المملكة المتحدة) /Almmlkĥ AlmtHdĥ/ ‘United Kingdom’. This tag is written thus #REDIRECTED[[المملكة المتحدة]].
3. **Disambiguation article:** These are used to list all the article titles that share ambiguities.

- **Links types:** There are two types of links in Wikipedia and they are described below:

1. **Non-piped links:** this type of links denotes that the display phrase of the link and the article’s title are the same. For example: [[London]].
2. **Piped links:** this type of link allows for the text that appears in the contextual data to be different from the actual article it refers to. For example: [[UK|United Kingdom]], where “UK” appears in the display text, while “United Kingdom” refers to the titles of the article.

Throughout this paper, the terms “link” and “link phrase” are used interchangeably to refer to the same thing.

- **Connectivity:** Used links, of any type, in the contextual data of any normal article, provide connectivity and thereby an underlying structure for Wikipedia; we are seeking to utilise to achieve our goal.

### 3 Transforming Arabic Wikipedia into a Fine-grained NE corpus and Gazetteer

In this section we present in detail the approach advised to automatically develop a tagged fine-grained named entity corpus and gazetteer based on Arabic Wikipedia.

#### 3.1 The Conciseness of the Approach

Our assumption regarding this work is as follows:

If we are able to classify Wikipedia articles into NE classes, we will then be able to map the resultant labelling back into contextualised linked phrases. This involves the following steps:

1. Defining a fine-grained taxonomy suitable to Wikipedia;
2. Classifying Arabic Wikipedia articles into a pre-defined set of fine-grained NE classes;
3. Mapping the results of the classification back to the linked phrases in the text;
4. Detecting successive mentions of NE that have not been associated with links, while taking into account the Arabic morphological variation of the NE phrase; and
5. Selecting sentences to be included in the final corpus.

#### 3.2 Defining Fine-grained Semantic NE Classes

Sekine et al. (2002) proposed a hierarchical named entity taxonomy that is very fine, with 150 subclasses. The methodology they used to construct semantic classes relies on analysing the named entities in a newswire corpus, in addition to analysing the answer type for a set of questions used in a Text Retrieval Conference TREC-QA task. WordNet noun hierarchy is also used to shape the classes further. Two years later, Sekine and Nobat (2004) added an extra 50 classes and decomposed some classes, such as “disease” and numeric expression respectively. Although the spectrum of classes is very wide, the specific descriptions and definitions for each class strives to avoid overlap and ambiguity, making it difficult to define. This taxonomy has been applied to both English and Japanese.

Some NLP applications, such as QA have designed their own named entity classes, based on the criteria they believe to be the most valuable. Harabagiu et al. (2003) developed a named entity recognition component in which one level consists of 20 defined fine grained classes. Knowing that factoid type questions require named entities, Li and Roth (2006) defined a fine grained taxonomy to answer certain types of questions. Although, their two layer taxonomy covered 50 fine grain classes of different types, some types were unrelated to named entities such as definition, description, manner and reason. Based on the same trend, Brunstein (2002) presented a two-level taxonomy in which 29 answer types are subdivided into 105 subtypes. Other researchers have adopted and used their taxonomy for named entity taxonomy (Nothman et al., 2008).

It is evident that there is no widely agreed fine grained taxonomy that can be directly adopted into Arabic; although ACE taxonomy is a reasonable choice

in the sense that it organises granularity into two layers, i.e. coarse and fine grained. In the evaluation of ACE (2008), the number of fine grain classes is 45. This taxonomy is designed in two levels of granularities and frequently used in the news-wire domain. Moreover, two-level taxonomy allows us to map a tagset into different traditional schemes easily, such as CoNLL or MUC.

Thus, ACE (2008) taxonomy was selected and because it is designed for a news-wire domain we applied some amendments to tailor it for use in a relatively open domain corpus, such as Wikipedia. For example, there are many articles in Wikipedia about people in different subclasses, such as scientists, athletes, artists, politicians, etc. These fine classes are not included in ACE, as it only involves three sub-classes: the individual, group and indeterminate. Another modification is performed; a new class called “Product” is added. This modified taxonomy is presented in Table 1.

Coarse-grained Classes	Fine-grained Classes
<i>PER: Person*</i>	<i>Politician*, Athlete*, Businessperson*, Artist*, Scientist*, Police*, Religious*, Engineer*, Group, Other*.</i>
ORG: Organisation	Government, Non-Governmental, Commercial, Educational, Media, Religious, Sports, Medical-Science, Entertainment.
LOC: Location	Address, Boundary, Water-Body, Celestial, Land-Region-Natural, Region-General, Region-International.
GPE: Geo-Political	Continent, Nation, State-or-Province, County-or-District, Population-Center, GPE-Cluster, Special.
FAC: Facility	Building-Grounds, Subarea-Facility, Path, Airport, Plant.
VEH: Vehicle	Land, Air, Water, Subarea-Vehicle, Unspecified.
WEA: Weapon	Blunt, Exploding, Sharp, Chemical, Biological, Shooting, Projectile, Nuclear, Unspecified.
<i>PRO:Product*</i>	<i>Book*, Movie*, Sound*, Hardware*, Software*, Food*, Drug*, Other*.</i>

Table 1: ACE (2008) modified taxonomy. The modified or added classes are represented with italics and asterisks

### 3.3 Wikipedia Document Classification

The aim of classifying Wikipedia articles is to produce a list of two tuples, like <article’s title, fine-grained NE tag>. The following sub sections describe the steps taken to achieve this goal.

#### 3.3.1 Fine-grained Document Annotation and Quality Evaluation

In order to classify Arabic Wikipedia articles into named entity classes, we manually annotated 4000 articles into two levels of granularity, i.e. coarse and fine grained, using the modified taxonomy shown in Table 1. Two Arabic natives were involved in the annotation process and the inter-annotator agreement between the annotators was calculated using Kappa Statistic (Carletta, 1996). Table 2 shows that the inter-annotator agreement was calculated for different sizes of documents, i.e. 500, 2000 and 4000. This revealed difficulties that might be encountered during the annotation process.

Level	<i>Kappa:</i> n=500	<i>Kappa:</i> n=2000	<i>Kappa:</i> n=4000
Coarse-grained	92	98	99
Fine-grained	80	95	97

Table 2: Inter-annotator agreement in coarse and fine grained levels

#### 3.3.2 Features Engineering and Representation

We developed our classification model relying on the set of features proposed by Alotaibi and Lee (2012) as these score 90% on the F-measure for coarse grained level. The features were:

1. **Simple Features (SF):** which represent the raw dataset as a simple bag of words without further processing.
2. **Filtered Features (FF):** involving removing the punctuation and symbols, filtering stop words and normalising digits.
3. **Language-dependent Features (LF):** represent the tokens in their stem form.
4. **Enhanced Language-dependent Features (ELF):** involving tokenising the sentence and assigning parts of speech for each token. This allows filtering of the dataset by involving only nouns (for instance) in the classifier.

In addition, we extended this set of features by extracting two more features:

1. **First paragraph:** Instead of just relying on the first sentence as in (Alotaibi and Lee, 2012), we identified useful features spread across the first paragraph.
2. **Bigram:** By using this feature, we aim to examine the effects of the collocation of tokens. Here we added the representation of a bigram while still preserving the unigram.

We represent the feature space using the term frequency-inverse document frequency (tf-idf).

### 3.3.3 Fine-grained Document Classification Results

The annotated dataset was divided into training and test at 80% and 20% respectively. We chose the Support Vector Machine (SVM) and Stochastic Gradient Descent (SGD) as a probabilistic model for the classifier. In each round of the classification, we tested one set of features and selected the one that performed best.

Table 3 shows the overall results for the fine-grained classification. There are three main findings. First, both classifiers tend to perform in a very similar way; therefore, in practice, use of either classifier to perform the final classification for the whole Wikipedia dataset will be expected to deliver very similar results. The second finding is that, the bigram features have little effect when different features are set. Finally, the best result for both classifiers was achieved using the  $ELF_{Uni}$  feature.

Features set	SVM			SGD		
	P	R	F	P	R	F
$SF_{Uni}$	0.78	0.79	0.78	0.78	0.79	0.78
$SF_{Uni+Bigram}$	0.80	0.81	0.80	0.80	0.81	0.79
$FF_{Uni}$	0.80	0.81	0.80	0.81	0.82	0.80
$FF_{Uni+Bigram}$	0.81	0.82	0.81	0.81	0.82	0.81
$LF_{Uni}$	0.77	0.78	0.77	0.78	0.79	0.78
$LF_{Uni+Bigram}$	0.79	0.80	0.79	0.79	0.80	0.79
$ELF_{Uni}$	0.82	0.83	0.82	0.82	0.83	0.82
$ELF_{Uni+Bigram}$	0.81	0.82	0.81	0.82	0.82	0.81

Table 3: The average fine-grained classification results when using SGD and SVM over different features sets where (tf-idf) is applied

### 3.4 Compiling the Corpus

Compilation of the final corpus was achieved according to the pipeline steps as follows:

1. Prepare and extract the features for all Arabic Wikipedia datasets, according to the method presented in Section 3.3.2;
2. Train an SVM classifier using the training dataset (4000 articles);
3. For each Wikipedia article, classify the article into the target fine-grained NE class;
4. Prepare final list of all articles' titles and their tags; and
5. Detect successive mentions of the named entity that have not been associated with the link:

As a convention, a linking phrase in the text of any Wikipedia article should only be assigned the first time it appears in context; successive mentions of the phrase appear with no link. Therefore, not all NE phrases are linked every time. Detecting successive mentions works by finding and matching

possible NE phrases in the text that share similarity, to a certain extent, with each phrase in the list of linked NE phrases. The main goal of this step is to augment the plain text with NE tags and to address some of the lexical and morphological variations that arise when a named entity is contextualised. For example, a named entity of (سعود الفيصل /sçwd Alfysl/ 'Saud Alfaisal') is expected to be repeated in context with either the first name (سعود /sçwd/ 'Saud') or the last name (الفيصل /Alfysl/ 'Alfaisal') or both together. This can also be difficult when prefixes are used. For example (و لسعود /wlsçwd/ 'and for Saud'). Therefore, we prepare for and match all the variations of prefixes that can be attached to the NE.

6. Produce the NE annotated corpus by selecting sentences to be included in the final corpus.

#### 3.4.1 To Which Extent to Select Sentences to be Involved in the Final Corpus?

We decided to compile two versions of the developed corpus. The first version is called "WikiFANE<sub>Whole</sub>", which means that we retrieved all the sentences from the articles. On the other hand, the second version, i.e. WikiFANE<sub>Selective</sub>, is compiled by selecting only the sentences, which have at least one named entity phrase. This creates a Wikipedia corpus that has as high a density of tags as possible.

In this paper and for evaluation purposes, we compiled the corpus for more than 2 million tokens as shown in Table 4. Meanwhile, this methodology allows all of Arabic Wikipedia to become a tagged fine-grained NE corpus. Moreover, both versions of this dataset were freely distributed to the research community<sup>2</sup>.

Corpus	# of sentences	# of tokens
WikiFANE <sub>Whole</sub>	76821	2,023,496
WikiFANE <sub>Selective</sub>	57126	2,021,177

Table 4: The total number of sentences and tokens for the compiled corpora

## 4 Introducing a Fine-grained Arabic NE Gazetteer

The process of classifying Wikipedia articles into NE classes provides the benefit of compiling a large Arabic NE Gazetteer at two levels of granularity. Based on our best knowledge, the only Arabic NE gazetteer currently available is that produced by Benajiba et al. (2007) covering only three traditional NE classes, i.e. PER, ORG and LOC. The size of this gazetteer

<sup>2</sup>The fine-grained Arabic NE corpora, i.e. WikiFANE<sub>Whole</sub> and WikiFANE<sub>Selective</sub> are freely available at <http://www.cs.bham.ac.uk/~f5a081/resources.html>



is 4132 entities. Table 5 compares the distribution between ANERgazet and WikiFANE<sub>Gazet</sub>. Due to the space limitation, we only present the coarse level distribution of WikiFANE<sub>Gazet</sub>. It is clearly shown that, WikiFANE<sub>Gazet</sub> has superiority in the sense of type and coverage. The gazetteer produced is freely available to the research community to use and extend<sup>3</sup>.

Class	ANERgazet	WikiFANE <sub>Gazet</sub>
PER	1920	30821
ORG	262	6664
LOC	1950	1424
GPE	NA	20785
FAC	NA	2182
VEH	NA	518
WEA	NA	274
PRO	NA	5624
Total	4132	68355

Table 5: The distribution of named entities for different gazetteers across coarse-grained NE classes

## 5 Evaluation and Results

To evaluate the fine-grained NE corpus and gazetteer produced, we conducted a set of thorough experiments. The aims of the evaluation were to answer the following questions:

- What is the quality of the corpus produced and the gazetteer in terms of annotation?
- How efficient is the NE classifier when used with WikiFANE<sub>Whole</sub> and WikiFANE<sub>Selective</sub> and tested over cross-domain and within-domain datasets?

### 5.1 Evaluating the Annotation Quality

The performance of document classification across all Wikipedia articles is crucial to avoid error propagation from the document classification stage when compiling the final version of the annotated corpus. Therefore, the first evaluation focused on this aspect. After classifying all articles to the target NE classes, we drew another 4000 articles, to be represented as a sample for all Wikipedia articles, and manually annotated them. The selection of the articles was made by selecting the first 4000 articles with identical glyphs to those used most frequently in other Wikipedia articles. This criteria ensured that the most frequent NE were classified properly with a minimum error rate. After this, we calculated the inter-annotation agreement between the manually annotated, gold-standard documents, and that classified based on step 3 in Section 3.4. Table 6 shows the result for both levels of granularity. The overall Kappa for the fine-grained level is 82.6% and this is

<sup>3</sup>The fine-grained Arabic NE gazetteer WikiFANE<sub>Gazet</sub> is freely available at <http://www.cs.bham.ac.uk/~fsa081/resources.html>

Level	Accuracy	Overall Kappa
Coarse-grained	85.8	84.02
Fine-grained	82.9	82.6

Table 6: Inter-annotation agreement between the classified articles and the gold-standard

Features
<i>Lexical features</i>
Current token
Two tokens before and after the current token
First and last three characters of the token
Length of the token
The tag of the previous token
<i>Morphological features</i>
Gender
Number
Person
<i>Syntactical features</i>
Part of speech
Base phare chunk
<i>External knowledge features</i>
The token appears in gazetteer

Table 7: The set of language dependent and independent features extracted to be used by the classifier

consistent with the results shown in Section 3.3.3. This gives the impression that, the error rate is at a minimum, even when performing the classification across all Wikipedia articles with small amounts of training data.

### 5.2 Evaluating the Corpus Developed by Learning NE Classifier

This evaluation was designed to evaluate the corpus developed by using it as training data to test it over cross-domain and within-domain datasets. Moreover, this assists evaluation of the efficiency of using gazetteer as external knowledge resource. We parsed the different datasets and tokenised the sentences using AMIRA (Diab, 2009) relying on the scheme (Conjunction + Preposition + Prefix). The concept behind using this tokenisation scheme is that, the notable sparseness issues regarding Arabic NE are caused by agglutination of the prefixes. In this scheme, we guaranteed that the named entities like (خالد /xAld/ ‘Khalid’) in the training data also refer to (ولخالد /wlxAld/ ‘and for Khalid’) in the test data. This happens by tokenising the words and splitting the prefixes, so the result will be three different tokens (و /w/ ‘and’), (ل /l/ ‘for’) and (خالد /xAld/ ‘Khalid’).

We extracted traditional sets of features at different levels; including lexical, morphological, syntactical and external knowledge. Table 7 summarises the features used where a window of five features are encoded in the classifier including the current position.

The following set of experiments was conducted relying on the Conditional Random Field (CRF) probabilistic model to perform the sequence labelling. In all the experiments, we divided the datasets into training and test at 80% and 20% respectively. We used the three metrics, precision, recall and F-measure, to evaluate the results.

### 5.2.1 Tags Distribution for the Gold-standard Newswire-based NE Corpora and WikiFANE

Different corpora have been used by researchers to develop NER. The first one is ANERcorp, which developed by Benajiba et al. (2007) and is freely accessible. It is a 150K news-wire based corpus tagged with CoNLL traditional coarse classes, i.e. PER, ORG, LOC and MISC. ACE produced two datasets named ACE 2004<sup>4</sup> and ACE 2005<sup>5</sup> which are subject to a costly licence. This prevents us using those corpora in the evaluation. However, ACE also produced a multilingual small corpus called REFLEX Entity Translation Training/DevTest (REFLEX for short), which consists of about 60K of tokens with two levels of classes. This is divided according to its origin into news-wire (NW), treebank (TB) and web blogs (WL). We used both the ANERcorp and the Arabic portion of REFLEX as gold-standard corpora to conduct the evaluation.

Table 8 shows the tag distribution for each corpus per class and the total per token and phrase. We use (NA) as an indication of no availability in the dataset. It is clearly shown that WikiFANE<sub>Selective</sub> has wider distributed tags compared with WikiFANE<sub>Whole</sub>.

### 5.2.2 Gazetteer Evaluation

Using gazetteer as an external knowledge source in NER helps to boost the performance of NER (Carreras et al., 2002). To evaluate the gazetteer produced, we learned the classifier by news-wire dataset one at a time. Each time, we evaluated the presence and absence of WikiFANE<sub>Gazet</sub>. Due to ANERcorp dataset being coarse-grain level, we decided to map the REFLEX dataset to the same scheme used by ANERcorp. In addition, we eliminated the MISC class used by ANERcorp because there is no direct equivalent in REFLEX. Three main points arose from this experiment. First, the F-measure increased by at least 2 points for all datasets, showing the overall positive effect of the developed gazetteer. Second, the recall metric clearly boosted the classifier enabling retrieval of more NE phrases than would be possible without WikiFANE<sub>Gazet</sub>. Third, the TB sub-dataset of REFLEX showed dramatic improvement in comparison with other datasets, because that TB dataset had comparatively less noise.

<sup>4</sup><http://www ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2005T09>

<sup>5</sup><http://www ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2006T06>

Coarse-grained classes	ANERcorp	REFLEX	WikiFANE <i>whole</i>	WikiFANE <i>selective</i>
PER	6505	2701	37009	84757
ORG	3454	2457	14133	35479
LOC	5069	220	7533	9886
MISC	1707	NA	NA	NA
GPE	NA	3472	70523	94099
FAC	NA	175	3651	6790
VEH	NA	47	269	482
WEA	NA	25	612	748
PRO	NA	NA	7717	14063
Total (token level)	16735	9097	141447	246304
Total (phrase level)	11175	7566	96562	142932

Table 8: The distribution of the coarse-grained NE tags across different corpora

Corpus	No Gazetteer			WikiFANE <sub>Gazet</sub>			
	P	R	F	P	R	F	
ANERcorp	87.13	69.27	77.18	87.86	72.34	79.35	
REFLEX	NW	88.51	69.37	77.78	88.21	72.79	79.76
	TB	79.09	70.16	74.36	89.20	76.61	82.43
	WL	83.78	62.23	71.41	84.69	66.61	74.57

Table 9: The comparison for using WikiFANE<sub>Gazet</sub> as external knowledge over news-wire dataset

### 5.2.3 Cross-domain Evaluation

The purpose of cross-domain evaluation is to train the classifier on a certain domain and then test this over different datasets with different domains or genres. The aim behind this experiment is to evaluate the effect when using WikiFANE<sub>Whole</sub> and WikiFANE<sub>Selective</sub> as training data versus news-wire domain datasets. This experiment helps to clarify the suitability of using WikiFANE as a relatively open domain corpus. It is evident from Table 10 that, self-training of ANERcorp and REFLEX produces the best performance. Meanwhile, there are some interesting findings. Even though REFLEX is a news-wire based corpus, its performance is dramatically lower when it is used as training dataset and tested over ANERcorp. This is also the case when training ANERcorp and testing it over REFLEX. This implies that, even within the same domain, news-wire, there is less generalisability for the current news-wire dataset across different datasets. Another interesting finding is that, the version of WikiFANE<sub>Selective</sub> performs better than WikiFANE<sub>Whole</sub> on different test sets, except for with ANERcorp. This might be because WikiFANE<sub>Selective</sub> has a greater tag density than WikiFANE<sub>Whole</sub>, which leads to more positive examples in the dataset.

Training	Testing											
	ANERcorp			REFLEX								
				NW			TB			WL		
	P	R	F	P	R	F	P	R	F	P	R	F
ANERcorp	87.86	72.34	79.35	80.60	58.38	67.71	79.31	64.92	71.40	74.23	52.55	61.54
REFLEX	73.57	50.07	59.59	88.21	72.79	79.76	89.20	76.61	82.43	84.69	66.61	74.57
WikiFANE <sub>Whole</sub>	81.53	43.10	56.39	71.43	37.84	49.47	84.11	51.21	63.66	71.43	36.50	48.31
WikiFANE <sub>Selective</sub>	88.10	37.52	52.62	86.99	42.16	56.80	86.49	51.61	64.65	84.43	37.59	52.02

Table 10: The result of cross-domain evaluation

Corpus		P	R	F
ANERcorp + WikiFANE <sub>Selective</sub>		90.40	58.21	70.81
REFLEX + WikiFANE <sub>Selective</sub>	NW	90.55	62.16	73.72
	TB	86.52	62.10	72.30
	WL	86.01	52.74	65.38

Table 11: The result of combining WikiFANE<sub>Selective</sub> with news-wire corpora

To elaborate more on cross-domain evaluation we evaluated the merging of WikiFANE<sub>Selective</sub>, since it performed best in the previous experiment, with both ANERcorp and REFLEX. The idea behind this experiment was to understand how the classifier performs when different domains and genera are combined together. The most notable findings, as shown in Table 11 are that, the recall metric shows a sharp drop in all datasets. However, the precision shows high scores, suggesting the Wikipedia corpus is strong in difference when compared with the news-wire domain.

### 5.2.4 Within-domain Evaluation

The traditional practice of learning NE classifier is to draw the training and test datasets from single domain. Therefore, we divided WikiFANE<sub>Whole</sub> and WikiFANE<sub>Selective</sub> into training and test for 80% and 20% respectively and then training the CRF classifier on WikiFANE<sub>Whole</sub> and WikiFANE<sub>Selective</sub> separately with and without the injection of the WikiFANE<sub>Gazet</sub> as an external knowledge source. Table 12 shows that, the use of WikiFANE<sub>Gazet</sub> creates a notable improvement across datasets by at least 3 points on the F-measure. In addition, WikiFANE<sub>Selective</sub> has a slightly superiority over WikiFANE<sub>Whole</sub> advising that both datasets are performing at a promising level of accuracy.

## 6 Related Work

A promising trend in the research is towards automatically developing an annotated NE corpus that extends beyond both traditional classes and the domain of newswire, in order to create novel resources. One of the earliest of these approaches was presented by An et al. (2003) in which the web was used to build a target corpus, using bootstrapping to build an anno-

tated NE corpus. A further approach utilises parallel corpora to build an NE corpus automatically. This relies on the suggestion that once one corpus is annotated then other parallel corpora can be easily annotated using projection. Ehrmann et al. (2011) developed multilingual NE corpora for English, French, Spanish, German and Czech. Similarly, Fu et al. (2011) developed a Chinese annotated NE corpus exploiting an English aligned corpus. The difference here is that the alignment is conducted between both corpora at the word-level.

Beyond the newswire-based corpora, Wikipedia becomes more attractive for different NLP tasks. Some researchers have exploited the unrestricted accessibility of Wikipedia to establish an automatic fully annotated NE corpus with different granularity; meanwhile others are merely focusing on partially utilising Wikipedia to achieve specific goals, such as developing a NE gazetteer (Attia et al., 2010) or classifying Wikipedia articles into NE semantic classes (Saleh et al., 2010).

Tkatchenko et al. (2011) expanded the classification into an 18 fine-grain taxonomy extracted from (BNN). To prepare training data for use in the classification stage, a small set of seeds is constructed, as undertaken by Nadeau et al. (2006), in which a semi-supervised bootstrapping approach was used to construct long lists of entities in different fine-grain NE classes from the web. After the list is constructed, the entities are then intersected with Wikipedia articles so as to classify each article according to its target class. Therefore, a set of 40 articles per fine-grain class was produced for use in training with the Naïve Bayes and Support Vector Machine (SVM). Several similar features have been selected (e.g. (Saleh et al., 2010; Dakka and Cucerzan, 2008)).

Instead of relying on machine learning, Richman and Schon Richman and Schon (2008) defined a set of heuristics involving using assigned category links to classify articles. Phrasal patterns for each semantic NE class were specified when a matching article was classified; alternatively the procedure searched the upper level of categories to find candidates. These articles are still classified according to traditional coarse grain classes.

Closely related to our work are attempts to build a completely annotated NE corpus free from human in-

Corpus	PER			ORG			LOC			Overall		
	P	R	F	P	R	F	P	R	F	P	R	F
WikiFANE <sub>Whole</sub> (no gaz)	93.15	85.41	89.11	93.69	89.34	91.46	83.39	66.81	74.19	88.51	76.18	81.88
WikiFANE <sub>Selective</sub> (no gaz)	92.82	85.80	89.17	93.41	88.83	91.06	81.76	72.24	76.70	86.92	78.62	82.56
WikiFANE <sub>Whole</sub>	97.35	88.61	92.78	97.74	93.10	95.36	84.58	70.37	76.83	91.10	79.62	84.98
WikiFANE <sub>Selective</sub>	96.37	88.75	92.40	96.12	91.73	93.87	82.55	75.73	78.99	88.77	81.86	85.18

Table 12: The result for within-domain evaluation

tervention. The first attempt to transform Wikipedia into an annotated NE corpus was made by Nothman et al. (2008); they assumed that many NEs are associated with Wikipedia inter-links, i.e. the hyperlinks associated with a phrase in contexts pointing to another article. Therefore, the procedure first identified NEs using heuristics to exploit capitalisation, and then the target articles were classified into NE semantic classes. A bootstrapping approach is then used to extract seeds from a set of 1300 articles. Two distinguishing features were extracted per article; i.e. the head noun for the category links and the head noun for the definitional sentence. The corpus produced covered 60 fine-grained classes in two layers. An alternative approach to the same data set is presented by Tardif et al. (2009), in which the classification relies on supervised machine learning. Like Dakka and Cucerzan (2008), both Naïve Bayes and the Support Vector Machine (SVM) have been used as statistical interfaces for the purpose of classification. A total of 2311 articles have been manually annotated and a combination of structured and unstructured features extracted.

The corpus produced by Nothman et al. (2008) has been thoroughly experimented with to evaluate the impact of its performance. Three different gold-standard corpora, i.e. MUC, CoNLL and BNN, were used for comparative purposes and separate models built for each corpus. The experiment showed that, when in conjunction with other gold-standard corpora the Wikipedia-based corpus could raise their performance; it also performs well for non-Wikipedia texts (Nothman et al., 2009).

## 7 Conclusion

We presented a methodology to develop a large size fine-grained named entity corpus and gazetteer using an automatic approach. This involved recruiting document classifications. Using this methodology, we produced constantly evolving NE resources that will exploit the yearly growth rate of Arabic Wikipedia.

The freely fine-grained NE corpus and gazetteer produced when used on their own are of a very promising quality and extend the scope of research beyond traditional NE tasks.

## References

- ACE. 2008. Ace (automatic content extraction) english annotation guidelines for entities, 06. [accessed 10 April 2013].
- Fahd Alotaibi and Mark Lee. 2012. Mapping Arabic Wikipedia into the named entities taxonomy. In *Proceedings of COLING 2012: Posters*, pages 43–52, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Joohee An, Seungwoo Lee, and Gary Geunbae Lee. 2003. Automatic acquisition of named entity tagged corpus from world wide web. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 165–168. Association for Computational Linguistics.
- Mohammed Attia, Antonio Toral, Lamia Tounsi, Monica Monachini, and Josef van Genabith. 2010. An automatically built named entity lexicon for arabic. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10)*, Valletta, Malta. European Language Resources Association (ELRA).
- Yassine Benajiba, Paolo Rosso, and José Miguel Benediruz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 4394 of *Lecture Notes in Computer Science*, pages 143–153. Springer Berlin / Heidelberg.
- Y. Benajiba, I. Zitouni, M. Diab, and P. Rosso. 2010. Arabic named entity recognition: Using features extracted from noisy data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 281–285, Uppsala, Sweden. Association for Computational Linguistics.
- Ada Brunstein. 2002. Annotation guidelines for answer types. LDC2005T33 [accessed 02 January 2012].
- Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, 22(2):249–254, June.
- Xavier Carreras, Lluís Marquez, and Lluís Padró. 2002. Named entity extraction using adaboost. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4. Association for Computational Linguistics.

- Wisam Dakka and Silviu Cucerzan. 2008. Augmenting wikipedia with named entity tags. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 545–552, Hyderabad, India. Asian Federation of Natural Language Processing.
- Mona Diab. 2009. Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*.
- Maud Ehrmann, Marco Turchi, and Ralf Steinberger. 2011. Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 118–124, Hissar, Bulgaria.
- Ruiji Fu, Bing Qin, and Ting Liu. 2011. Generating chinese named entity data from a parallel corpus. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 264–272, Chiang Mai, Thailand. Asian Federation of Natural Language Processing (AFNLP).
- Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On arabic transliteration. In *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 15–22. Springer Netherlands.
- Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, John Williams, and Jeremy Bensley. 2003. Answer mining by combining extraction techniques with abductive reasoning. In *Proceedings of 12th Text Retrieval Conference*, volume 2003, pages 375–382. NIST.
- Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(03):229–249.
- David Nadeau, Peter D. Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. *Advances in Artificial Intelligence*, pages 266–277.
- Elisa Noguera, Antonio Toral, Fernando Llopis, and Rafael Muñoz. 2005. Reducing question answering input data using named entity recognition. In *Text, Speech and Dialogue*, pages 428–434. Springer.
- Joel Nothman, James Curran, and Tara Murphy. 2008. Transforming wikipedia into named entity training data. In *Proceedings of the Australian Language Technology Association Workshop*, pages 124–132, Hobart, Australia. ALTA.
- Joel Nothman, Tara Murphy, and James Curran. 2009. Analysing wikipedia and gold-standard corpora for ner training. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620, Athens, Greece. Association for Computational Linguistics.
- Alexander Richman and Patrick Schon. 2008. Mining wiki resources for multilingual named entity recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1–9, Columbus, Ohio, USA. Association for Computational Linguistics.
- Iman Saleh, Kareem Darwish, and Aly Fahmy. 2010. Classifying wikipedia articles into ne’s using svm’s with threshold adjustment. In *Proceedings of the 2010 Named Entities Workshop*, pages 85–92, Uppsala, Sweden. Association for Computational Linguistics.
- Satoshi Sekine and Chikashi Nobat. 2004. Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the 4th International Conference on Language Resources And Evaluation*, pages 1977–1980, Lisbon, Portugal. ELRA.
- Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of the third International Conference on Language Resources and Evaluation*, volume 2, Las Palmas, Spain. ELRA.
- Khaled Shaalan and Hafsa Raza. 2007. Person name entity recognition for arabic. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 17–24, Prague, Czech Republic. Association for Computational Linguistics.
- Sam Tardif, James Curran, and Tara Murphy. 2009. Improved text categorisation for wikipedia named entities. In *Australasian Language Technology Association Workshop 2009*, pages 104–108, Sydney, Australia.
- Maksim Tkatchenko, Alexander Ulanov, and Andrey Simanovsky. 2011. Classifying wikipedia entities into fine-grained classes. In *Data Engineering Workshops (ICDEW), 2011 IEEE 27th International Conference on*, pages 212–217. IEEE.
- Wikipedia. 2013. The statistic of arabic wikipedia, 05. [accessed 10 May 2013].

# Ranking Translation Candidates Acquired from Comparable Corpora

Rima Harastani and Béatrice Daille and Emmanuel Morin

LINA UMR CNRS 6241 - University of Nantes

2 rue de la Houssinière, BP 92208

44322 Nantes, France

{rima.harastani,beatrice.daille,emmanuel.morin}@univ-nantes.fr

## Abstract

Domain-specific bilingual lexicons extracted from domain-specific comparable corpora provide for one term a list of ranked translation candidates. This study proposes to re-rank these translation candidates. We suggest that a term and its translation appear in comparable sentences that can be extracted from domain-specific comparable corpora. For a source term and a list of translation candidates, we propose a method to identify and align the best source and target sentences that contain the term and its translation candidates. We report results with two language pairs (French-English and French-German) using domain-specific comparable corpora. Our method significantly improves the top 1, top 5 and top 10 precisions of a domain-specific bilingual lexicon, and thus, provides a better user-oriented results.

## 1 Introduction

Comparable corpora have been the subject of interest for extracting bilingual lexicons by several researchers (Rapp, 1995; Fung and Mckeown, 1997; Rapp, 1999; Koehn and Knight, 2002; Morin et al., 2008; Bouamor et al., 2013, among others). Rapp (1995) was the first to suggest that if a word  $A$  co-occurs frequently with another word  $B$  in one language, then the translation of  $A$  and the translation of  $B$  should co-occur frequently in another language. Approaches emerging from (Rapp, 1995) make different assumptions to extract bilingual lexicon from comparable corpora. However, they are all based on the assumption that a translation pair shares some similar context in comparable corpora. We refer to such approaches that depend on co-occurrences of

words to extract a bilingual lexicon by *distributional approaches*. Results obtained from distributional approaches vary according to many parameters. For example, one of the parameters that impacts the performance of distributional approaches is the way the context of a word is defined. Various approaches defined contexts differently: windows (Rapp, 1999), sentences or paragraphs (Fung and Mckeown, 1997), or by taking into consideration syntax dependencies based on POS tags (Gamallo, 2007). However, the most common way the context of a word is defined is by choosing words within windows centered around the word (Laroche and Langlais, 2010), usually of small sizes (e.g. a window of size 3 is used by Rapp (1999)).

Domain-specific comparable corpora have been used for bilingual terminology extraction. These corpora are of modest sizes since large domain-specific corpora are not available for many domains (Morin et al., 2008). As a matter of fact, distributional approaches perform best with large comparable corpora, and thus they often give lower precisions when applied to domain-specific comparable corpora (Chiao and Zweigenbaum, 2002).

The goal of our work is to find translations of terms in domain-specific comparable corpora. Taking a list of ranked translation candidates (provided by a distributional method) for a term, we aim to improve the ranking of the correct translations that are not ranked first in the list. Obviously, the more translation candidates for a term are considered, the more correct translations are found. For example, Rapp (1999) obtains a precision of 72% when only the first translation candidate is considered correct. However, he reports an 89% precision when the first 10 translation candidates are provided as translations for a word.

This study proposes to take the best translation candidates provided by a distributional approach,

and tries to re-rank them in order to improve the top 1, top 5 and top 10 precisions. We suggest that a source term and its correct translation appear in comparable sentences. Comparable sentences are sentences that share parallel data (e.g. word overlap, long matched sequences, bilingual compound nouns). We proceed by first extracting sentences for a source term, as well as sentences for each of its provided translation candidates. For each translation pair (i.e. source term and a translation candidate), each extracted source sentence is aligned with at most one of the extracted sentences for the translation candidate. The aligned sentences are used to re-rank the translation candidates of the source term.

Besides being used by our approach to re-rank translations, comparable sentences that contain a term and its translation in corpora are promising, as they may be useful examples to a user or a human translator that needs to verify a translation pair.

In Section 2, we present our approach and assumptions. In Section 3, we describe our method to extract sentences that best represent a term in corpora. In Section 4, we explain a method to score a sentence containing a term with a sentence containing its translation candidate. We evaluate our approach in Section 5 on two domain-specific corpora for the French-English and French-German language pairs, and report improvements in the top 1, top 5, and top 10 precisions. We conclude in Section 6.

## 2 Assumptions and Approach

A term may appear in several contexts, but some can be more interesting and more informative than others. In Table 1, an example of two sentences in which the term “tumor” appears is given. These sentences were extracted from an English corpus related to the domain of “Breast Cancer”. Sentence (A) is considered to be more informative and more representative of the context of “tumor” than sentence (B). It also contains terms that are highly related to the “Breast Cancer” subject (e.g. chemotherapy, histological).

Our assumption is that the best context (represented by sentences) can be extracted for a term as well as for its translation candidates, and that these extracted sentences can be aligned in order to re-rank the translation candidates of the term.

After obtaining some candidate translations for

(A)	<b>Chemotherapy</b> was also administered to patients with smaller primary <u>tumors</u> with <b>histological</b> grade 2 or 3 or with negative hormone receptors.
(B)	The size of any captured image corresponding to the <u>tumor</u> was estimated.

Table 1: Sentence (A) and (B) containing the term “tumor”

a term by applying a distributional method, we score a source term ( $t_s$ ) with its target translation candidate ( $t_t$ ) as follows: we first extract the  $n$  best sentences that contain  $t_s$  in the source corpus as well as the  $n$  best sentences that contain its translation candidate in the target corpus. Then, we align each of the best sentences of  $t_s$  with at most a sentence of  $t_t$  using a method that depends on lexical similarity. Finally, the translation pair ( $t_s, t_t$ ) is scored according to the scores of the aligned sentences between  $t_s$  and  $t_t$ . The scoring method is illustrated in Figure 1. We combine the resulting score with its initial score that is provided by a distributional method. Combined scores are then used to re-rank translation candidates of the specific term.

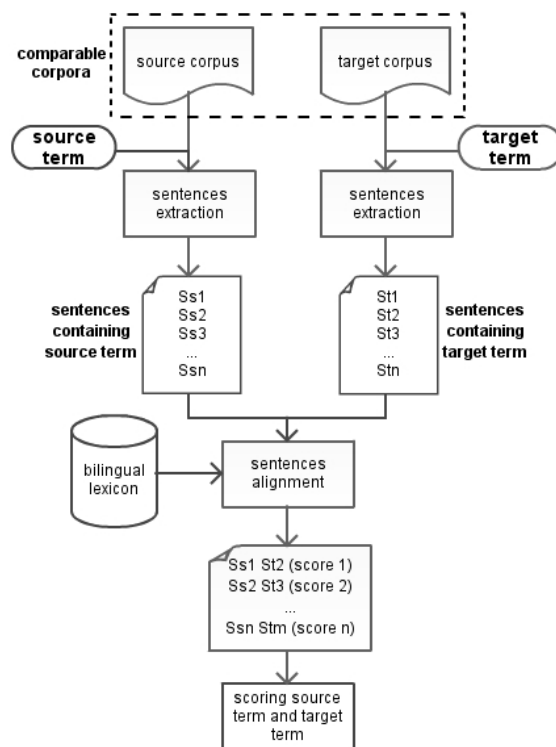


Figure 1: Method to score a translation pair (source term and target term)

Parallel sentence (or fragment) extraction from comparable corpora has received the attention of a number of researchers (Fung and Cheung, 2004; Munteanu and Marcu, 2005; Munteanu and Marcu, 2006; Smith et al., 2010; Hunsicker et al., 2012, among others), to enrich parallel text used by statistical machine translation (SMT) systems. They conducted experiments with large corpora (mainly news stories) which were noisy parallel, comparable (contain topic alignments or articles published in similar circumstances), or very non-parallel (Fung and Cheung, 2004). Usually, these approaches perform document-level alignments before extracting parallel sentences. The domain-specific corpora we use contain few documents (ranging from 38 to 262 documents for each corpus) and no parallel sentences. Furthermore, they are of modest size (about 0.3 M to 0.5 M words), so even if there were some parallel fragments, this phenomenon would be rare. Nevertheless, we assume that some features used in state-of-the-art parallel sentence extraction methods can be used to identify comparable sentences that contain a translation pair.

Our goal is not to extract parallel sentences, but rather we need to find, for a translation pair, bilingual sentences that are comparable. For example, consider that we need to score the correct translation pair (FR<sup>1</sup> clinique, EN<sup>2</sup> clinical), and that we have two sentences, the first contains “clinique” and the second contains “clinical” (see Figure 2). The two sentences are not parallel, however, they both contain the following information: a clinical examination detects the size of a tumor. Finding this kind of comparability in sentences would help in increasing the score of correct translation pairs.

### 3 Best Sentences Extraction for a Term

For a term ( $t$ ), we aim to extract the  $n$  best sentences that represent its context in the corpus. We suggest that sentences that best represent  $t$  contain words that are: (a) strongly associated with  $t$  in the corpus, (b) highly specific to the domain of the corpus. A word in a sentence containing  $t$  is scored by means of two measures: association and domain specificity, that are presented in the following.

1. Association with  $t$ : word associations are computed according to log-likelihood scores

<sup>1</sup>FR signifies French

<sup>2</sup>EN signifies English

that are based on the co-occurrences of words in a window of size ( $s=7$ ) around  $t$ . The top ( $m=30$ ) associated words and their scores with  $t$  are denoted by  $v_m$  (context vector of  $t$  of size  $m$ ). The association between a word ( $w$ ) and  $t$  is computed from occurrences that are resumed in the contingency table (see Table 2), where  $\text{occ}(t,w)$  is the number of occurrences of  $t$  and  $w$ , and  $\neg w$  signifies all words except  $w$ .

	<b>w</b>	<b><math>\neg w</math></b>
<b>t</b>	a=occ( $t,w$ )	b=occ( $t,\neg w$ )
<b><math>\neg t</math></b>	c=occ( $\neg t,w$ )	d=occ( $\neg t,\neg w$ )

Table 2: Contingency table for  $t$  and  $w$

The log-likelihood association measure is computed as follows:

$$\begin{aligned} \text{association}(t, w) = & a \log(a) + b \log(b) \\ & + c \log(c) + d \log(d) + (N) \log(N) \\ & - (a + b) \log(a + b) - (a + c) \log(a + c) \\ & - (b + d) \log(b + d) - (c + d) \log(c + d) \end{aligned} \quad (1)$$

where  $N = a + b + c + d$ . The association between  $w$  and  $t$  is then divided by the biggest association score obtained with  $t$  to have a score  $\in [0,1]$ .

2. Domain specificity: the specificity of a word is its relative frequency in the domain-specific corpus ( $dc = \{w_1, w_2, \dots, w_n\}$ ) divided by its relative frequency in a general language corpus ( $gc = \{w'_1, w'_2, \dots, w'_m\}$ ), it is defined in (Khurshid et al., 1994) as follows:

$$ds(w) = \frac{rvf_{dc}(w)}{rvf_{gc}(w)} \quad (2)$$

where  $rvf_{dc} = \frac{freq_{dc}(w)}{\sum_{w_i \in dc} freq_{dc}(w_i)}$  is the relative frequency in the specific corpus,  $rvf_{gc}(w) = \frac{freq_{gc}(w)}{\sum_{w'_i \in gc} freq_{gc}(w'_i)}$  is the relative frequency in the general corpus, and  $freq$  signifies frequency. The specificity of a term is normalized by being divided by the value of the biggest specificity in the corpus.

To extract the  $n$  best sentences for term  $t$ , we give a score to each sentence  $S$  that contains  $t$  and words



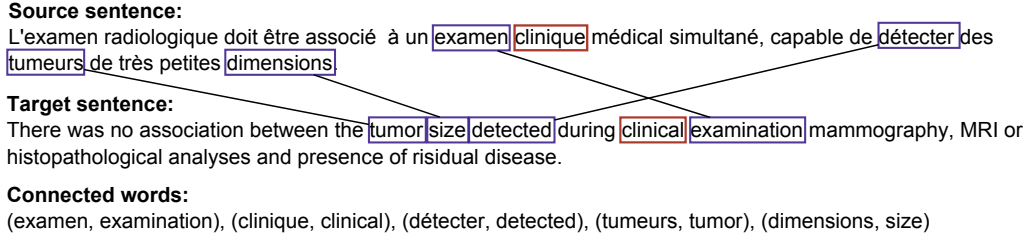


Figure 2: Example of source and target sentences that contain the translation pair (FR clinique and EN clinical)

$w_1, w_2, \dots, w_n$  as follows:

$$score(S) = \sum_{i=1}^n \left( ds(w_i) + association_{(if w_i \in v_m)}(w_i, t) \right) \quad (3)$$

We discard any sentence with a length of less than 5 words (after removing the stop words). All sentences containing  $t$  are then ranked according to their scores. For a translation pair  $(t_s, t_t)$ , the  $n$  best sentences for  $t_s$  as well as for  $t_t$  are extracted following the method explained above.

The next step consists of aligning the  $n$  best sentences of a source term  $t_s$  with  $n$  best sentences of each of its proposed translations.

#### 4 Sentences Alignment for Translation Pairs

We suggest that if a source term ( $t_s$ ) is translated by a target term ( $t_t$ ), then they must share some comparable sentences. The more a translation pair shares sentences with high comparability, the higher its score should be.

The ratio between the lengths of two comparable sentences should be less than 2, following (Munteanu and Marcu, 2005). We also suppose that the overlap between two comparable sentences should be greater than 3 (including the translation pair). Like previous works on extracting parallel sentences from comparable corpora, our approach depends mostly on lexical information between sentences by using a bilingual lexicon.

Suppose that we have a source sentence  $S_s = \{w_1, w_2, t_s, \dots, w_n\}$ <sup>3</sup> and a target sentence  $S_t = \{w'_1, w'_2, t_t, \dots, w'_n\}$ <sup>4</sup> (after removing the stop words), with a set of possible connected words

<sup>3</sup> $t_s$  could be at any position in  $S_s$   
<sup>4</sup> $t_t$  could be at any position in  $S_t$

$M = \{(w_1, w'_1), (w_2, w'_2), \dots, (w_n, w'_n)\}$  obtained using a bilingual dictionary. An optimal alignment  $A$  (each word in the sentence  $S_s$  is connected to at most one word in the sentence  $S_t$ ) is estimated according to a linear function.

Taking the optimal alignment  $A$ , feature functions (where each  $\in [0, 1]$ ) are utilized to compute a score between the two sentences.

1. The cosine similarity between the two sentences (Fung and Cheung, 2004) penalized by the number of unconnected words: each word in  $S_s$  (respectively  $S_t$ ) is weighted by its score in the context vector  $v_m$  (respectively  $v'_m$ ) of  $t_s$  (respectively  $t_t$ ). If a word is missing from the context vector, it would be associated a fixed minimal weight. The first feature function is defined as follows:

$$f_1(S_{t_s}, S_{t_t}) = \frac{\text{cosine}(S_{t_s}, S_{t_t})}{|\text{UnConnectedWords}|} \quad (4)$$

where  $|\text{UnConnectedWords}|$  is the number of unconnected words between the two sentences.

2. Positions of connected words in the source sentence (target sentence respectively) in comparison to the position of source term (target term respectively): the nearer the connected words are from the term  $t$  in the sentence, the greater the score of this feature function will be. Besides, we suppose that for two connected words  $(w_i, w'_i)$ , the distance between  $w_i$  and  $t_s$  should be close to the distance between  $w'_i$  and  $t_t$ . The positions distance is defined as follows:

$$pos_{\text{distance}}(S_{t_s}, S_{t_t}) = \sum_{w_i, w'_i \in A} \frac{(pos_s + pos_t + |pos_s - pos_t|)}{(|S_{t_s}| + |S_{t_t}| + |S_{t_s} - S_{t_t}|)} \quad (5)$$

where  $pos_s = |pos(w_i) - pos(t_s)|$  and  $pos_t = |pos(w'_i) - pos(t_t)|$ .

The  $pos_{distance}$  is then divided by  $|A|$  to be normalized. The positions similarity is computed as follows:

$$f_2(S_{t_s}, S_{t_t}) = 1 - \frac{pos_{distance}}{|A|} \quad (6)$$

3. Longest contiguous span: it is defined by (Munteanu and Marcu, 2005) as being the longest “pair of substrings in which the words in one substring are connected only to words in the other substring”. We assume that the length of a span must be greater than 2. The longest span is divided by the length of the smaller sentence, then:

$$f_3(S_{t_s}, S_{t_t}) = \frac{\text{span}(S_{t_s}, S_{t_t})}{\min(|S_{t_s}|, |S_{t_t}|)} \quad (7)$$

4. Number of connected bi-grams: this feature function is defined as the number of found connected bi-grams divided by the number of connected words in  $A$ , then:

$$f_4(S_{t_s}, S_{t_t}) = \frac{\text{bi-grams}(S_{t_s}, S_{t_t})}{|A|} \quad (8)$$

The optimal alignment  $A$  is the alignment that minimizes the squared Euclidean distance between the two sentence vectors and the  $pos_{distance}$ . Indeed, we choose this minimization function for a matter of optimization.

We follow (Hunsicker et al., 2012) in considering the final score between a sentence pair as the weighted sum of all feature functions, such as the following:

$$\text{score}(S_s, S_t) = \sum_{i=1}^4 (w_i * f_i(S_{t_s}, S_{t_t})) \quad (9)$$

where  $\sum_{i=1}^4 (w_i) = 1$ .

Contrary to previous works that use parallel corpora to train their models and define the weights of feature functions, we define the weights by guesswork. This is because we do not have an annotated parallel corpora. Nevertheless, this should not have a significant impact on our results since our goal is not to extract parallel sentences.

#### 4.1 Reranking translation pairs

For a translation pair  $(t_s, t_t)$ , each sentence of the  $n$  best representing sentences of  $t_s$  is aligned with at

most one of the  $n$  best representing sentences of  $t_t$ . A target sentence can be aligned to multiple source sentences. The score between the translation pair is the average of the scores of the sentence alignments. We refer to this procedure as the sentence alignment method.

The re-ranking is done by combining the score obtained by the sentence alignment method for a translation pair with its initial score that is obtained by a distributional method. The scores are combined by the weighted geometric mean.

## 5 Evaluation

We first need to extract translations for a list of domain-specific terms in comparable corpora. In order to do this, we pre-process corpora and align terms with the free tool TermSuite<sup>5</sup> (Rocheteau and Daille, 2011). The distributional method that is implemented in TermSuite is the one described in (Rapp, 1999). TermSuite provides a chosen number of translations for a term. Translations are ranked according to the scores provided by the distributional method. We try to enhance the top candidate translations of each reference source term by applying our re-ranking method.

### 5.1 Data

To carry out the distributional approach with TermSuite, we need comparable corpora, bilingual dictionaries, and a list of source reference terms to translate. We need the same resources to perform experiments with our method as well as general language monolingual corpora.

- Comparable corpora: we carry out experiments with comparable corpora in two different domains and two language pairs French-English and French-German. The first are medical corpora in the sub-domain of *breast cancer*, these contain approximately 0.37 M to 0.5 M words for each language. The second corpora belong to the renewable energy domain, more specifically, to the sub-domain of *wind energy*, and contain about 0.3 M to 0.35 M words for each language. Breast Cancer corpora were collected from an online medical portal, while Wind Energy corpora have been crawled using Babouk crawler (Groc, 2011). Both corpora have been collected using some seed terms and contain no

<sup>5</sup>This tool is available on <http://code.google.com/p/ttc-project/>

parallel sentences. Table 3 resumes the sizes of monolingual parts of corpora.

Language	Breast Cancer	Wind Energy
French	531,240	313,943
English	528,428	314,549
German	378,474	358,602

Table 3: Sizes in number of words of corpora for each language and for each domain

- Bilingual dictionaries: general language bilingual dictionaries<sup>6</sup> for the French-English and French-German language pairs were obtained. The French-English dictionary contains 145,542 single-word entries and the French-German dictionary contains 118,776 single-word entries.
- General language corpora: for each language, a general language corpus is obtained and used in computing specificities of words to the domain-specific corpora. These contain 12003, 3903 and 44365 unique single words for French, English and German respectively.
- Reference lists: we have built a list of reference single-word terms (SWTs) for each corpora and for each language pair. Each source term in the list is domain-specific with a frequency greater than 5 in the source corpus and has been manually aligned with one golden translation that exists in the target corpus. For Breast Cancer corpora, for each language pair we built a list that contains 122 translation pairs. As for Wind Energy corpora, for each language pair we built a list that includes 96 translation pairs.

## 5.2 Experimental Settings

For the sentence alignment method, we manually define the same parameters for Breast Cancer and Wind Energy corpora. For each term and each translation candidate, we extract the 70 best sentences, where sentences that have the same score are ranked at the same position. However, we take a maximum of 200 sentences for a term. If a term is less frequent than 70 in the corpus, we extract all the sentences that include this term. We do not

<sup>6</sup>The dictionaries were obtained from [http://catalog.elra.info/product\\_info.php?products\\_id=666](http://catalog.elra.info/product_info.php?products_id=666) and [http://catalog.elra.info/product\\_info.php?products\\_id=668](http://catalog.elra.info/product_info.php?products_id=668)

extract a large number of sentences for a term because the alignment process will be computationally expensive, besides, our assumption is that if a translation pair is valid, then its best representative sentences are comparable. When extracting sentences for a term, we discard any sentence with a length of less than 5 words (after removing the stop words). A sentence is supposed to be simply delimited by punctuation marks (“?”, “!”, “.”). We point out that the words, in a sentence containing a term  $t$ , that are used in computing the score of this sentence and as context for  $t$  are the words appearing at maximum in a window of size  $n=20$  around  $t$  (10 words or less appearing before  $t$  in the sentence, and 10 words or less appearing after  $t$  in the sentence, after removing the stop words).

To score a translation pair by aligning its sentences (see equation 9), the biggest weight is set to 0.4 and is attributed to the first feature function (see equation 4). The remainder of weights are set equally to 0.2. When combining the scores of the distributional and the sentence alignment methods by the weighted geometric mean, the weight of the first is set to 0.3, and the weight of the second is set to 0.7.

## 5.3 Evaluation Measures

The precision of a bilingual lexicon is computed at different levels after taking several  $n$  best translations for each term (top 1, top 5, etc.). The precision is the number of the correct translations found divided by the number of source terms in the reference list.

The Mean Reciprocal Rank (MRR) is also used to evaluate the obtained results. The reciprocal rank for a given source term is the multiplicative inverse of the rank of the first correct target translation. The mean reciprocal rank is the average of the reciprocal ranks of the aligned source reference terms. MRR values are between 0 and 1, where higher values indicate a better performance of the system.

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (10)$$

where  $|Q|$  is the number of source terms to be aligned. If a the correct translation of a term has not been found, then its corresponding “ $\frac{1}{\text{rank}_i}$ ” is equal to 0.

## 5.4 Experiments

The results of the distributional approach (baseline) with the language pairs and two corpora are given in Table 4 (P1 signifies the precision when 1 translation candidate is provided for a term). We notice that the results on Breast Cancer corpora are better than those obtained with Wind Energy. This may be justified by the fact that Wind Energy corpora are of smaller sizes and less technical.

The results are also significantly better with the French-English language pair than with the French-German language pair. In fact, domain-specific corpora contain many terms that are compound nouns. In the German language, many compound nouns may be written as single units (e.g. German term “Produktionsstandort” is translated into French by “site de production”). Therefore, the distributional approach may consider such German terms as one word when computing co-occurrences. One way to overcome this problem would be to perform splitting before applying the distributional approach (Macherey et al., 2011).

To analyze the results obtained by the distributional method in more depth, we measured the comparability of Wind Energy corpora for the different language pairs, using the comparability measure presented by Li et al. (2011). For the French-English corpora, we obtained a comparability value of 0.81. As for the French-German corpora, we obtained a comparability value of 0.70. This implies that our French-German corpora are less comparable than the French-English corpora, and partly justifies the reason behind obtaining worse results with the French-German pair using the distributional method.

	Breast Cancer		Wind Energy	
	FR-EN	FR-GR	FR-EN	FR-GR
<b>P1</b>	26.22%	9.16%	16.66%	3.12%
<b>P5</b>	45.08%	18.85%	38.54%	9.37%
<b>P10</b>	53.27%	26.22%	45.83%	10.41%
<b>P15</b>	59.01%	29.50%	50.00%	12.50%
<b>P20</b>	60.65%	31.96%	57.29%	14.58%
<b>P25</b>	61.47%	32.78%	59.37%	14.58%

Table 4: Results obtained with distributional method (baseline). EN-FR signifies English-French, and FR-GR signifies French-German.

In order to improve these results, especially the top 1, top 5 and top 10 precisions, we try to re-rank

the translation candidates for each source term by combining their initial scores with the scores obtained from aligning their sentences.

Let us suppose that for a source term  $t_s$ , we want to re-rank its top 5 translation candidates  $L_{top5}=\{t_{t_1},t_{t_2},t_{t_3},t_{t_4},t_{t_5}\}$  provided by the distributional method. Following the approach presented in Section 3, we extract the best ranked sentences for  $t_s$ . We do the same for each translation candidate in  $L_{top5}$ . Then, for each translation pair (e.g.  $t_s$  and  $t_{t_1}$ ) we try to align each sentence that was extracted for  $t_s$  with one sentence that shares the highest score with it among the sentences extracted for  $t_{t_1}$ , using the approach described in Section 4. A source sentence can be aligned with at most one target sentence and is assigned a score (which is equal to 0 if the sentence is not aligned). The score between  $t_s$  and  $t_{t_1}$  is the average of the scores of the alignments.

Following the above explained procedure, we take the best  $n=20$  translation candidates proposed by the distributional method for each term and re-rank the translation candidates. This evaluation strategy is denoted by RR1 in Tables 5 and 6 which resume the obtained results on our corpora with two language pairs. For example, using the French-English Breast Cancer list, we find that re-ranking the top 20 translation candidates provided for each source term improved the top 1 precision by approximately 5%. Moreover, before re-ranking, 43.24% of the correct translations found in the top 20 results were ranked at the 1<sup>st</sup> position, after re-ranking, this percentage increases to 52.70%. Which means that the re-ranking has significantly improved the ranks of the correct translations. An improvement of approximately 6% in the top 1 precision is obtained when using 20 translation candidates to re-rank the results obtained with the French-English Wind Energy list. However, fewer improvements were obtained with the French-German language pair as there were not many correct translations in the first 20 translations provided for each term by the distributional method.

While performing experiments, we have noticed that re-ranking the first 5 translation candidates for each term may increase the top 1 precision more than if we, for example, re-ranked the first 20 translation candidates for each term. For that, we have decided to follow a different strategy (denoted by RR2) for re-ranking translations. To de-

	Breast Cancer			Wind Energy		
	Baseline	RR1	RR2	Baseline	RR1	RR2
<b>P1</b>	26.22%	31.96%	<b>35.24%</b>	16.66%	<b>23.95%</b>	22.91%
<b>P5</b>	45.08%	<b>52.45%</b>	<b>52.45%</b>	38.54%	<b>45.83%</b>	44.79%
<b>P10</b>	53.27%	<b>57.37%</b>	<b>57.37%</b>	45.83%	48.95%	<b>52.08%</b>
<b>MRR</b>	0.338	0.396	<b>0.419</b>	0.249	<b>0.324</b>	0.319

Table 5: Results obtained on both Breast Cancer and Wind Energy French-English Corpora

	Breast Cancer			Wind Energy		
	Baseline	RR1	RR2	Baseline	RR1	RR2
<b>P1</b>	9.16%	<b>11.47%</b>	<b>11.47%</b>	3.12%	<b>7.29%</b>	5.20%
<b>P5</b>	18.85%	<b>21.31%</b>	<b>21.31%</b>	9.37%	<b>10.41%</b>	<b>10.41%</b>
<b>P10</b>	26.22%	<b>27.04%</b>	<b>27.04%</b>	10.41%	<b>13.51%</b>	<b>13.51%</b>
<b>MRR</b>	0.139	0.160	<b>0.162</b>	0.051	<b>0.088</b>	0.075

Table 6: Results obtained on both Breast Cancer and Wind Energy French-German Corpora

termine which translation candidate will be ranked at the  $n$  (starting from 1) position for a term, we first re-rank the top  $m = (\text{round}(2(n-1)+5))$  to the nearest multiple of 5) translations proposed for each term. The translation candidate at position 1 will have the position  $n$  in the new ranked list and it will not be further re-ranked. Then, we determine the translation candidate that will be ranked at the position  $(n+1)$  in the new ranked list. We repeat this process until obtaining 10 translation candidates for each term in the new ranked list.

For example, taking a list of translation candidates provided for a term: to determine which translation candidate will be ranked at the first position, we re-rank the list of top 5 ( $L_{top5}$ ) translation candidates provided for the term, we put the translation now ranked in the first position in a list we name  $L_{taken}$ . To determine which translation candidate will be in the second position, we re-rank the list ( $L_{top5} - L_{taken}$ ) and add the translation ranked in the first position to  $L_{taken}$ . Now to determine which translation will be ranked in the third position, we re-rank the (list of top 10 -  $L_{taken}$ ), and put the translation ranked in the first position in  $L_{taken}$ , and so on. Results obtained using this strategy are presented in Tables 5 and 6 (under RR2).

RR2 strategy gave better top 1 precision and MRR than RR1 with French-English Breast Cancer corpora, and better top 10 precision with French-English Wind Energy corpora. RR1 strategy gave better MRR on Wind Energy corpora. In general, the results of the two strategies were

comparable. This means that RR1 gave stable improvements when re-ranking a list of 20 candidates for each term. Both RR1 and RR2 significantly improved the baseline results for French-English and French-German language pairs.

## 6 Conclusion

In this paper, we proposed a method to re-rank the top translation candidates acquired by a distributional method from comparable corpora. We assumed that some sentences are more representative of a term than others, and that a term and its correct translation share comparable sentences that can be extracted from comparable corpora. We suggested aligning sentences that best represent a term with sentences that best represent its translation candidates to re-rank these translation candidates. Our experiments showed improvements in precision and MRR measures for two language pairs and two domains.

Our re-ranking method was tested with SWTs, and we aim to further evaluate it with multi-word terms (MWTs). Moreover, best aligned sentences for a term and its translation candidates can also be proposed for a user-oriented evaluation to see whether the aligned sentences can help in validating a translation pair.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable remarks. This work was supported by the French National Research Agency under grant ANR-12-CORD-0020.

## References

- Dhouha Bouamor, Nasredine Semmar, and Pierre Zweigenbaum. 2013. Context vector disambiguation for bilingual lexicon extraction from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Short Papers)*, volume 2 of *ACL '13*, pages 759–764, Sofia, Bulgaria.
- Yun-Chuang Chiao and Pierre Zweigenbaum. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th international conference on Computational linguistics*, volume 2 of *COLING '02*, pages 1–5.
- Pascale Fung and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 57–63, Barcelona, Spain.
- Pascale Fung and Kathleen Mckeown. 1997. Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora*, pages 192–202.
- Pablo Gamallo. 2007. Learning bilingual lexicons from comparable english and spanish corpora. In *Machine Translation Summit 2007*, pages 191–198.
- Clément De Groc. 2011. Babouk : Focused Web Crawling for Corpus Compilation and Automatic Terminology Extraction. In *The IEEE/WIC/ACM International Conferences on Web Intelligence*, pages 497–498, Lyon, France.
- Sabine Hunsicker, Radu Ion, and Dan Stefanescu. 2012. Hybrid parallel sentence mining from comparable corpora. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, EAMT '12, Trento, Italy.
- Ahmad Khurshid, Davies Andrea, Fulford Heather, and Rogers Margaret. 1994. What is a term? the semi-automatic extraction of terms from text. In *Translation Studies: An Interdiscipline*, John Benjamins Publishing Company, Amsterdam, pages 267–278.
- Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of ACL Workshop on Unsupervised Lexical Acquisition*, pages 9–16.
- Audrey Laroche and Philippe Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 617–625.
- Bo Li, Eric Gaussier, and Akiko Aizawa. 2011. Clustering comparable corpora for bilingual lexicon extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2 of *HLT '11*, pages 473–478, Portland, Oregon.
- Klaus Macherey, Andrew M. Dai, David Talbot, Ashok C. Popat, and Franz Och. 2011. Language-independent compound splitting with morphological operations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1 of *HLT '11*, pages 1395–1404, Portland, Oregon.
- Emmanuel Morin, Béatrice Daille, Koichi Takeuchi, and Kyo Kageura. 2008. Brains, not brawn: The use of smart comparable corpora in bilingual terminology mining. *ACM Trans. Speech Lang. Process.*, 7(1):1–23.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31:477–504.
- Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88, Sydney, Australia.
- Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 320–322, Cambridge, Massachusetts.
- Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 519–526, College Park, Maryland.
- Jerome Rocheteau and Béatrice Daille. 2011. TTC TermSuite: A UIMA Application for Multilingual Terminology extraction from Comparable Corpora. In *the 5th International Joint Conference on Natural Language Processing*, IJCNLP '11, pages 9–12, Chiang Mai, Thailand.
- Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 403–411.

# Using the Semantic-Syntactic Interface for Reliable Arabic Modality Annotation

**Rania Al-Sabbagh**

Department of Linguistics  
University of Illinois at Urbana-Champaign  
alsabba1@illinois.edu

**Jana Diesner**

School of Library and Information Science  
University of Illinois at Urbana-Champaign  
jdiesner@illinois.edu

**Roxana Girju**

Department of Linguistics  
University of Illinois at Urbana-Champaign  
girju@illinois.edu

## Abstract

We introduce a novel modality scheme where triggers are words and phrases that convey modality meanings and subcategorize for clauses and verbal phrases. This semantic-syntactic working definition of modality enables us to design practical and replicable annotation guidelines and procedures that alleviate some shortcomings of current purely semantic modality annotation schemes and yield high inter-annotator agreement rates. We use this scheme to annotate a tweet-based Arabic corpus for modality information. This novel language resource, being the first, initiates NLP research on Arabic modality.

## 1 Introduction

Modality is the grammaticalized expression of the "speaker's subjective attitudes" (Bybee et al., 1994:176) and "psychological stances" (Mitchell and al-Hassan, 1994:7) towards propositions and events and their factuality status. In NLP applications and domains, modality is considered as one linguistic means to convey and detect attitudes and opinions (Wiebe et al., 2005; Abdul-Mageed and Diab, 2012), commitments and beliefs (Diab et al., 2009), power relations (Prabhakaran et al., 2012), uncertainties and speculations (Szarvas et al., 2008; Matsuyoshi et al., 2010).

We herein present the first work on Arabic modality annotation, which is part of a larger research project to (1) automatically identify modality triggers (i.e. words and phrases conveying modality meanings), holders (i.e. modality experiencers) and scopes (i.e. the range of linguistic constituents modified by the modality triggers), and (2) automatically detect power re-

lations among participants in the social network of Twitter by using this modality information.

Despite solid work on Arabic modality in theoretical linguistics (Mitchell and al-Hassan, 1994; Brustad, 2000; Moshref, 2012), there are no Arabic corpora annotated for modality, not even the widely used Penn Arabic Treebank. However, there is a plethora of work and annotated corpora for modality in other languages, including English (Saurí et al, 2006; Baker et al., 2010; Prabhakaran et al., 2012; Rubinstein et al., 2013), Portuguese (Hendrickx et al., 2010; Avila and Mello, 2013), Japanese (Matsuyoshi et al. 2010) and Chinese (Cui and Chi, 2013).

Arabic modality annotation involves multiple challenges. First, the paradigm of Arabic modality triggers is complex as it includes auxiliaries, lexical verbs, nominals and particles - like many other languages as well. Second, triggers can be lexically and/or semantically ambiguous: a lexically-ambiguous trigger is a word or phrase that may or may not convey a modality meaning based on context. A semantically-ambiguous trigger is a word or phrase that may convey two or more modality meanings. Third, implicit scopes are common in Arabic and annotators have to be made aware of them. Fourth, Arabic word order flexibility allows triggers - especially adverbials - to occur in the scope's initial, medial or final positions, which makes it challenging for annotators to identify scope spans. Finally, modality scopes are not necessarily adjacent to their triggers, which furthermore complicates the detection of their spans.

The tweets genre on which we work poses an additional challenge due to language variation. We select a random sample of Arabic tweets from the YADAC corpus (Al-Sabbagh and Girju, 2012) posted in Egypt during the first six months

of the 2011 revolution. All selected tweets are about the political situation at that time. Tweets are not only in the Egyptian Arabic (EA) dialect, but also in Modern Standard Arabic (MSA), especially tweets from press agencies and celebrities. Therefore, our annotation scheme has to work on both MSA and EA modality.

Arabic and the tweets genre are not the only original aspects of this paper. We present a novel linguistically-motivated annotation scheme with a semantic-syntactic working definition of modality triggers as *words and phrases that convey modality meanings and subcategorize for clauses and verbal phrases*. Modality meanings are based on Palmer's (1989, 2001) cross-lingual typology of modality, which is proven valid for both MSA and EA (Mitchell and al-Hassan, 1994; Brustad, 2000; Moshref, 2012).

The semantic-syntactic interface between modality triggers and their scopes depicted in our definition is well-established in theoretical linguistics for Arabic (Moshref, 2012) and for English (Jackendoff, 1972; Brennan, 1993; Butler, 2003). Semantics and syntax work simultaneously such that semantics guarantees not to tag all words and phrases that subcategorize for clauses and verbal phrases as modality triggers. Syntax guarantees not to tag words and phrases that share some semantic meanings with modality triggers, but do not subcategorize for clauses and verbal phrases, as modality triggers.

Subcategorization frames of modality triggers are sporadically mentioned in the NLP literature on modality annotation. For English, Saurí et al. (2006) stated in their definition of event modality triggers that "they subcategorize for a that, gerundive or infinitival clause, but also an NP headed by event denoting nouns" (p.334). For Portuguese, Hendrickx et al. (2010) stated that "in the majority of the cases, the target [i.e. scope] is a subordinate clause or a verbal phrase ... in some cases, also main clauses can be targets [i.e. scopes]" (p. 1810). However, no prior work integrates the semantic-syntactic interface into a modality annotation scheme.

Our conceptualization and implementation of this semantic-syntactic interface provide annotators with practical annotation guidelines that yield highly-reliable results, as shown herein. Furthermore, they define modality in terms of concrete syntactic features which we use in our future work for the automatic identification of triggers and their scopes.

The rest of the paper is structured as follows: section 2 briefly reviews related work. Section 3

gives details about Arabic modality and its complexities. Section 4 presents our data, annotation guidelines and procedures. Section 5 reports the annotation results. Finally, we conclude with future work in Section 6.

## 2 Related Work

Recent work on modality annotation focuses on English, Portuguese, Japanese and Chinese. Baker et al. (2010) used an idiosyncratic categorization of English modality that distinguished 8 semantic meanings: requirement, permissive, success, effort, intension, ability, want and belief. They defined each type as a pattern of the form  $H$  (modal)  $P$  where  $H$  is the sentence's agent and  $P$  is the proposition (e.g.  $H$  permits [ $P$  to be true/false]). They obtained an average inter-annotator agreement rate of 0.82. The error analysis of their modality tagger showed that errors resulted primarily from the triggers' lexical ambiguity.

Prabhakaran et al. (2012) focused on 5 semantic meanings of English modality, and used the same  $HP$  patterns as Baker et al. (2010) for annotation guidelines. They reported an inter-annotator agreement rate of 0.95. Their modality tagger yielded a 0.44  $F_1$  score against a gold-standard and 0.79 and 0.91  $F_1$  scores against different testing sets from their crowdsourced data.

Rubinstein et al. (2013) used a more standardized typology of English modality that entailed (1) priority modality divided into bouletic, teleological and deontic triggers; and (2) non-priority modality divided into epistemic, circumstantial and ability triggers. Their purely semantic annotation scheme returned an alpha reliability score of 0.89 only when collapsing the subtypes of priority and non-priority triggers. The scheme yielded an alpha reliability score of 0.65 for scope span annotation.

Cui and Chi (2013) applied Rubinstein et al.'s (2013) scheme for modality annotation to the Penn Chinese Treebank. They obtained a reliability score of 0.94 for triggers' annotation using the collapsed binary typology of modality triggers as priority vs. non-priority. Their error analysis reported vagueness in the annotation guidelines as one disagreement factor.

The lack of previous NLP work on Arabic modality, modality annotation in tweets and syntactically-guided modality annotation schemes render direct comparisons to our work impossible. Yet, the two main distinguishing factors of our work are: (1) to guarantee the replicability of



our study, we avoid idiosyncratic typologies of modality that were used in some previous work; and (2) to better guide our annotators, we use practical guidelines that rely on both semantics and syntax rather than semantics only as in previous annotation schemes.

### 3 Arabic Linguistic Modality

#### 3.1 Background

Among multiple typologies of modality, Palmer's (1989, 2001) was validated for both MSA and EA in theoretical linguistics (Mitchell and al-Hassan, 1994; Brustad, 2000; Moshref, 2012).

Palmer distinguishes two main classes of modality: propositional and event. Propositional modality is concerned with the speaker's attitude to the truth-value of a proposition, and includes:

- Epistemic modality, which expresses the speaker's judgment about the factual status of the proposition as well as the speaker's opinion and attitude towards that proposition.
- Evidential modality, which indicates the evidence the speaker has for his or her judgment or opinion. Evidence can be reported as in hearsay and quotes or sensory.

Event modality refers to events that are not actualized but are merely potential, and includes:

- Deontic modality, which relates to obligations and permissions that emanate from an external source, and commissives, which originate from an internal source as speakers lay an obligation on themselves for a potential event.
- Dynamic modality, which relates to ability, willingness and wishes.

#### 3.2 Challenges

The challenges of Arabic modality annotation are attributed to (1) the complexity of the Arabic modality paradigm, (2) the lexical and semantic ambiguity of Arabic modality triggers, (3) implicit scopes, (4) word order flexibility and (5) potential long dependencies between triggers and their scopes.

The paradigm of Arabic modality triggers includes a large set of auxiliaries, lexical verbs, nominals and particles. Except for auxiliaries, adverbs and some particles, all modality triggers inflect for gender, number, person, tense, aspect and mood. Furthermore, generic modality patterns such as \* *mn Al-\* >n* (it is \* that), where \* is typically an adjective (e.g. *من المهم أن*

*mn Almhm >n* (it's important that)), are common.

Modality triggers can be lexically and/or semantically ambiguous. The noun *zmAn* is one example of a lexically-ambiguous trigger because in 1 it is an epistemic with a clause scope. Yet, in 2 it is a non-modal standing for *era*.

1. *مبارك لسه بيحكنا عندها خيرة*  
*lw smEnA klAm AlnAs Ally EndhA xbrp kAn zmAn mbArk lsh byHkmnA*

If we'd listened to the elite, Mubarak would have been still ruling us.

2. *الحكم الفردي انتهى*  
*zmAn AlHkm Alfrdy AnthY xlAS*

The **era** of individual rulers has come to an end.

The MSA particle *lAbd* is one example of semantically-ambiguous triggers because in 3 it is an epistemic with a clause scope; whereas in 4 it is an obligative with a verbal-phrase scope.

3. *أنه تذكر صدام حسين وهو يستلقي على سريريه الطبي*

*<n mbArk lAbd w>nh tzkr SdAm Hsyn whw ystlqy Ely sryrh AlTby fy qAEp AlmHkmp*

**It must be that Mubarak remembered Saddam Hussein** as he was lying on his medical bed in the court.

4. *يوضع*  
*lAbd An ywDE mbArk fy Alsjn*

Mubarak **must be put in jail**.

Implicit modality scopes are common in Arabic and come in different realizations. In 5, the scope of the permissive *nsmH* (allow) is the deictic \**lk* (that) which refers to the clause *يهان المصري* *In yhAn AlmSrywn* (Egyptians won't be humiliated). That is, the scope of *nsmH* is actually a clause.

5. *يهان مصري*  
*In yhAn AlmSrywn. In nsmH b\*lk.*

Egyptians won't be humiliated. We won't **allow** it.

In 6, the abilitives *AErf* (can) and *EArf* (can) share the same verbal-phrase scope of *A\$wf* (see). To avoid redundancy, the speaker elides the scope of the second abilitive - *AErf* (can) - and does not replace it with any deictic expression. Thus *AErf* modifies an implicit verbal-phrase scope.

6. *m\$ EArf A\$wf HAjp lmA AErf Hklmk*

I can't see anything. When I **can**, I'll call you.

On the surface level, the obligative *lAZm* (must) in 7 is followed by the noun phrase *a real reaction against military trials*. Yet, on a deeper level, the tweet is the short version of *we must*

(take) a real reaction against military trials. This means that *lAz*m has an implicit verbal-phrase scope.

العسكرية ( ) .7

*lAz*m (*naxd*) *mwqf bjd Dd AlmHAKmAt AlEskryp*

We **must** (take) a real reaction against military trials

Word order flexibility allows for some modality triggers - especially adverbials - to occur before, after or in the middle of their scope(s).

Long dependencies between modality triggers and their scope(s) are the last challenge with Arabic modality annotation. The obligative *ATlb* (ask; require) in 8 subcategorizes for a complement clause, which starts 9 words later (affixes excluded).

.8 من عناصر الاجهزة الامنية المصرية المتخفية في ملابس مدنية انهم يتفرجو عشان سلوكهم ولبسهم ونظرتهم مهروشة أوي

*ATlb mn EnASr AlAjhzp Al>mnyp AlmSryp Almtxfyp fy mlAbs mdnyp Anhm ytfrijw Ely kwnAn E\$An slwkhm wlbshw wnZrthm mhrw\$P >wy*

I **ask** Egyptian security individuals disguising in civil outfits to watch Conan because their behavior, outfit and looks are ridiculously revealed.

## 4 Arabic Modality Annotation

### 4.1 Corpus Encoding and Description

We randomly selected a corpus of 1,704 raw tweets (33,349 tokens and 11,013 unique types) from the YADAC corpus (Al-Sabbagh and Girju, 2012). The considered time span ranges from January 25, 2011 to June 30, 2011. All tweets were posted in Egypt by ordinary individuals, celebrities (e.g. politicians, actors, singers, TV hosts), and the press (e.g. newspapers, TV stations, NGOs, election campaigns).

The corpus includes tweets in both MSA and EA because press users always post in MSA, while celebrities and ordinary individuals frequently switch between MSA and EA. Based on our manual annotation of user types, we have 1,318 tweets posted by individuals, 369 tweets by celebrities and 17 tweets by the press.

### 4.2 Annotators and Annotation Units

Two EA native speakers performed the annotation. Being linguistics students, they can be assumed to master MSA. They were given a one-hour video tutorial covering the annotation guidelines and procedures in Sections 4.3 and 4.4, respectively, followed by a 30-minute workshop dedicated to training and discussion.

Each annotator is required to label each (1) modality trigger, (2) its semantic meaning, (3) its scope type(s), and (4) its scope span(s). We keep

holder annotation for future work as it poses additional challenges.

### 4.3 Annotation Guidelines

Our core annotation guidelines are summarized in the semantic-syntactic working definition of modality given in Section 1. We define modality triggers as words and phrases that (1) convey a modality meaning from Palmer's (1989, 2001) typology, (2) and subcategorize for clauses and verbal phrases; representing propositions and events, respectively. We also give the annotators a number of supplementary guidelines.

Annotators have to label each trigger and its scope(s). Multiple triggers may have the same scope as in 9 where the two epistemic triggers *EArf* (I know) and *mt>kd* (I'm sure) share the clause scope of *that Mubarak won't be executed*.

حيثعدم .9

*AnA EArf wmt>kd An mbArk m\$ HytEdm*

I **know** and I'm **sure** that Mubarak won't be executed.

Annotators have to label all the scopes of the modality trigger for type and then identify their spans. In 10, the obligative *lAz*m (must) modifies three verbal-phrase scopes linked by the coordinating conjunction *w* (and).

مانسكتش عليه نجيبه من جوده يحصلش تاني .10

*AHnA lAz m ntAbE AlmwDwE wmAnskt\$ Elyh wnjybh mn jdwrh E\$An mA yHSI\$ tAny*

We **must follow up** with this, **not ignore** it and **investigate** it well so it won't happen again.

Finally, annotators have to retrieve implicit scopes whether they are referred to in-text or using their own real-world knowledge.

### 4.4 Annotation Procedure

Annotation proceeded in four stages. For **Stage 0**, we used our novel, manually-built, large-scale Arabic Modality Lexicon (AML) to automatically pre-highlight candidate modality triggers. AML was built in three steps:

- First, we manually generated the person, gender, number, tense, mood and aspect inflections as well as the present and past participle derivations of 276 lemmas compiled from Mitchell and al-Hassan (1994), Brustad (2000) and Moshref (2012).
- Second, we added a list of triggers including particles, adverbs and multi-word generic expressions that do not inflect for person, gender, number, tense, mood and aspect.
- Finally, we labeled each entry for an English

<entry id="997" token="منهياً" trans="mthy>ly" gloss="I think" ambiguity="NA" dialect="EA" semClass="epistemic" features="NA" </entry>
<entry id="2032" token=" " trans="mn AlmHtm" gloss="it's essential that" ambiguity="NA" dialect="MSA" semClass="obligative" features="MWE" </entry>
<entry id="3423" token=" " trans="Ejz" gloss="failed to" ambiguity="lexical" dialect="MSA/EA" semClass="abilitive" features="inherentlyNeg,Quasi" </entry>

Table 1: An expert from the Arabic Modality Lexicon (AML)

gloss, ambiguities {lexical, semantic, both, NA}, dialects {MSA, EA, both}, modality semantic meaning and special features {quasi; inherently-negative, multi-word expression} as in Table 1.

Currently, AML has 7,584 entries, with the statistical distributions in Table 2. Despite the large size of AML, annotators were instructed to add any words or phrases that match our working semantic-syntactic definition of modality.

Semantic meanings		Ambiguity	
Epistemic	3,144	Lexical	2,363
Sensory	134	Semantic	155
Reported	427	Lexical/Semantic	116
Obligative	1,091	Unambiguous	4,950
Permissive	815		
Commissive	132		
Abilitive	957		
Volitive	884		
Dialects		Special Features	
MSA	2,268	Quasi	777
EA	3,100	Inherently-Neg.	788
MSA/EA	2,216	MWE	276

Table 2: AML statistics

For **Stage 1**, annotators labeled each pre-highlighted modality trigger for its modality semantic meaning. We defined modality semantic annotation as a synonymy judgment task where the annotators, given a number of synsets, had to decide to which synset the pre-highlighted trigger belongs. We used 8 synsets; each of which featured one modality semantic meaning from Palmer's (1989; 2001) typology. The average size of the synsets is 15 words/phrases to represent different shades of meaning. Yet, due to space limitations, we only included sample synsets in Table 3. To avoid fatigue, disinterest, and distraction effects, we used counterbalancing and prompted the annotators to provide their own synonym(s) for the pre-highlighted candidate trigger if none of the given synsets seemed synonymous.

For **Stage 2**, annotators labeled the syntactic type of the linguistic constituents modified by the pre-highlighted modality trigger (i.e. scope type) where applicable. Annotators had to choose whether the modified constituent was a clause, a verbal phrase, or another type of constituency (e.g. a noun phrase, an adjectival phrase). Once

the clause or the verbal phrase option was selected, annotators were prompted to extract that clause or verbal phrase.

<b>Epistemic</b> (opinion, conclusion, possibility)	- - - - في رأيه - - - -
<b>Evidential</b> (reported)	- - - - - - - -
<b>Evidential</b> (sensory)	شاهد بعينه - سمع بنفسه -
<b>Obligative</b> (and necessity)	- - - - يجب - - - -
<b>Permissive</b> (and prohibitive)	- - - - - نهى عن ... - - - -
<b>Commissive</b>	- - - - تعهد - عاهد ... - - - -
<b>Abilitive</b> (incapability)	- - - - تسنى له - - - -
<b>Volitive</b>	- - - - قد النية على - - - -

Table 3: Sample synsets used for modality semantic annotation

For **Stage 3**, we automatically extracted the triggers that followed our semantic-syntactic working definition of modality. That is, triggers labeled as synonymous to one of the synsets in Table 3 AND as modifying a clause or a verbal phrase. In this stage, instances such as *AfkrwA* in 11 were automatically excluded: although it modifies a complement clause, it means *remember* which is a non-modality meaning.

11. نمشي تحقيق مطالبنا بيبيقى

*AfkrwA* <n kl mrp bnm\$y tHqyq mTAlbnA bybqY <SEb  
**Remember** that every time we leave, it becomes harder to achieve our demands!

Similarly, instances such as *>ElnwA* (they announced) in 12 were automatically filtered out. It is synonymous with the evidential reported synset, yet it modifies a prepositional phrase.

12. أخيرا عن أسماء المعتقلين في السجن الحربي

*>ElnwA* >xyrA En >smA' AlmEtqlyn fy Alsjn AlHrby  
They finally **announced** the names of all prisoners in the military jail.

Instances such as *qAdr* (able to) in 13 were automatically admitted as valid modality triggers. It belongs to the dynamic abilitive synset, and modifies the verbal phrase *>n ySnE* (to make).

13. أن يصنع مصيره بيده

*Al\$Eb AlmSry qAdr >n ySnE mSyrh bydh*

The Egyptian people **are able to** make their own destiny.

Our annotation procedure pinpoints the efficiency and applicability of each dimension of our definition of modality. These guidelines also speed up the annotation process and increase annotation reliability because they provide annotators with practical and concrete prompts, and elicit well-structured answers that can be automatically converted into the modality annotation profiles described in Section 4.5.

#### 4.5 Modality Annotation Profiles

Twitter terms of services prohibit redistributing raw tweet texts. Thus at the end of the annotation process, a profile was built for each tweet with its user name, tweet ID, and modality-related information. Associated software is to be given to help reconstruct tweets using their IDs. Although at the time of writing this paper all tweet IDs are still active, there is a potential of degradation if users delete their tweets or make their accounts private. This does not affect the modality-related profile, however, the complete tweet text will not be available. Modality-related information presents chunks of the tweet texts that represent the trigger word/phrase and the scope clause/verbal phrase. Thus we assume that we are not violating the terms of services.

Triggers are marked with 4-character labels. The first character is *T* for Trigger. The second two characters indicate the semantic meaning of the trigger {*Ep*: epistemic, *Rp*: reported, *Sn*: sensory, *Ob*: obligative, *Pr*: permissive, *Cm*: commissive, *Ab*: abilitive, *Vl*: volitive}. The fourth character is an index to indicate whether the trigger is the 1<sup>st</sup>, 2<sup>nd</sup> and so on in the tweet and to relate the trigger to its scope(s).

Scopes are marked with 3-character labels. The first is *S* for Scope. The second represents the syntactic type of the scope - {*C*: clause, *P*: verbal phrase}. The last is an index matching that of its trigger. Table 4 shows the modality annotation profiles for examples 1 and 9, respectively.

---

```

user="alaa"
tweet_id="71857458888458240"
[[ ("zmAn", "TEP1"),
  ("mbArk lsh byHkmnA", "SC1") ]]

user="eAiNet"
tweet_id="46316910177697792"
[[ ("EArf", "TEP1"),
  ("An mbArk m$ HytEdm", "SC1")],
  [("mt>kd", "TEP2"),
  ("An mbArk m$ HytEdm", "SC2")]]

```

---

Table 4: Example modality annotation profiles

## 5 Annotation Results

### 5.1 Inter-Annotator Agreement Rates and Disagreement Factors

AML pre-highlighted 2,892 candidate triggers in our 1,704 tweets. We used the kappa statistics to measure the Inter-Annotator Agreement (IAA) rates for:

- **Modality semantic annotation:** this labels each candidate trigger as synonymous to one of the synsets in Table 3 featuring Palmer's (1989; 2001) typology.
- **Modality syntactic annotation:** this includes (1) **identifying the scope type** as to whether it is a clause, a verbal phrase, or none; and (2) **identifying the scope span** in terms of the beginning and the end of each scope.

Our macro kappa IAA rate for modality semantic annotation is 0.899 (Table 5). It is hard to measure if this rate is significantly higher than rates reported in the literature of modality semantic annotation because direct comparison with prior work is not possible as explained in Section 2. Yet, one point to highlight is that we do not use a collapsed typology of modality semantic meanings as in Rubenstein et al. (2013) and Cui and Chi (2013), who both collapsed modality semantic meanings into two major classes only: priority vs. non-priority.

	Kappa	Percent Agreement
Semantic annotation	0.899	0.918
Scope type	0.846	0.902
Scope span	0.929	0.973

Table 5: Macro kappa inter-annotator agreement rates for modality semantic and syntactic annotations

We attribute our high IAA rate for modality semantic annotation to: (1) the large-scale AML, which provides annotators with an extensive list of candidate triggers; and (2) using synonymy judgments to give annotators practical, self-evident annotation prompts instead of subjective guidelines, defining modality triggers as expressions of alternative states in which the world could be.

There are, however, two limitations to using synonymy judgments for modality semantic annotation. First, the quality of the annotation relies on the quality of the used synsets. It is important to select unambiguous triggers to represent the modality semantic meanings in different contexts. This is because triggers interact with other linguistic features such as modification, negation and grammatical mood. Second, to better guide

the annotators, especially when working on a morphologically-rich language such as Arabic, it is better to have the synset members inflected for the same person, gender, number, tense, mood and aspect as the candidate trigger.

It took us three iterations of annotations - each one with two different annotators - to come up with the best final synsets used in this paper. This process is time and labor consuming. Yet, once the synsets have been created, the annotation process is fast and replicable with a potential to be crowdsourced. We will examine this option in future work.

**Highly-ambiguous lemmas** are the first disagreement factor for modality semantic annotation. Epistemic lemmas such as *\$Af* (saw), *Erf* (knew), *fhm* (understood), *Sdq* (believed) and *qAl* (said) among others have multiple meanings of which one or more might be modality-related. This explains why most of the disagreement scores in Table 6 are between modality and non-modality meanings (i.e. NA).

	<i>Ep</i>	<i>Rp</i>	<i>Sn</i>	<i>Ob</i>	<i>Pr</i>	<i>Cm</i>	<i>Ab</i>	<i>VI</i>	<i>NA</i>
<i>Ep</i>	610	10	3	1	9	0	9	0	33
<i>Rp</i>	7	261	0	13	0	0	0	0	20
<i>Sn</i>	2	0	192	0	0	0	0	0	5
<i>Ob</i>	0	8	0	299	3	0	1	1	0
<i>Pr</i>	0	0	0	1	93	0	6	0	0
<i>Cm</i>	0	0	0	0	0	7	0	0	0
<i>Ab</i>	0	0	0	0	0	0	124	0	0
<i>VI</i>	0	0	0	0	0	0	0	267	1
<i>NA</i>	98	6	0	0	0	0	0	0	802

Table 6: Confusion matrix for semantic annotation

For one annotator *nfhm* in 14 is synonymous to *n\$RH* (explain) and thus does not belong to any of the modality synsets in Table 3. For the other annotator, this trigger still means *explain*, but not as in explaining factual information, but as in making people adopt a specific point of view or a belief. Thus it is synonymous to *njElhm y&mnwn >n* (make them believe that) and is an epistemic trigger.

14. نفهم إن المجلس العسكري حاجة والجيش حاجة

*lAbd nfhm AlnAs <n Almjls AlEskry HAjp wAljy\$ Hajp*

We must **explain to** people that the Supreme Council of Armed Forces is one thing and the army is another.

The same lemma may also have more than one closely-related modality meaning; (i.e. it is semantically-ambiguous). For one annotator *byqwl* in 15 is an evidential reported trigger meaning *is saying*; whereas it is an epistemic trigger meaning *is thinking* for the other.

15. قفشوا ناس معاهم سلاح وطلعوهم بره الميدان عشان بس اللي بيقول  
إننا بلطجية يعرف إنه غلطان

*qf\$wA nAs mEAhm slAH wTIEwhm brh AlmydAn E\$an Ally byqwl <nnA blTjyp yErf <nh glTAn*

They arrested some people with guns and kicked them out of the square so that those **saying we're thugs** realize that they are wrong.

**Modality triggers not included in AML** are the second disagreement factor for modality semantic annotation. A total of 168 triggers were identified as new; 85 of which were agreed upon by both annotators. For future modality annotation, agreed-upon new triggers will be added to AML and controversial ones are to be examined by experts prior to inclusion.

The macro kappa IAA rate for scope type identification is 0.846 according to Table 5. Main factors of disagreement are:

- **Clauses vs. verbal phrases:** in some contexts, triggers such as *mmkn* (may, it's possible that) and *Drwry* (must; it is necessary that) can be understood either as auxiliaries subcategorizing for verbal phrases or as adjuncts subcategorizing for clauses. Thus *Drwry nnzl nqwl l>* can be either *we must protest and say no or it's necessary that we protest and say no.*
- **Implicit scope recovery:** implicit scopes with deictic expressions or in-text reference were easy to retrieve unlike implicature-based scopes. For instance, *AHnA Sdqna xTABh* (we believed his speech) was perceived by one annotator as *we believed what he said in his speech was true.* Thus the annotator selected the clause option for the scope type. The other annotator did not see such an implicature and thus selected *NA*, meaning that the scope is unrecognizable or is neither a clause nor a verbal phrase.

The macro kappa IAA rate for scope span recognition in Table 5 is 0.929 which is quite high. We attribute this to the simplicity of the tweet genre, which entails short sentences (140 characters or less) and a writing style that resembles short telegraphic notes more than formal and lengthy sentences. Interjections, adjuncts and subordinate conjunctions are the main reasons for disagreement. In 16, one annotator ends the span of the clause-based scope before *El\$An* (so that); while the other includes the entire sentence into the scope span. We will add clearer guidelines for when interjections, adjuncts and subordinate clauses should be considered into the scope span in future work.

16. (علشان اللي في الشارع يحسوا ان فيه حد جنبيهم)

*lAzM AlnAs tnzl (El\$An Ally fy Al\$ArE yHswA An fyh Hd jnbhm)*

People **must** go to protests (so that those already protesting won't feel as if left alone).

As we implemented Stage 3 of our annotation procedure, we sought triggers that adhere to our semantic-syntactic working definition of modality. Triggers labeled as conveying a modality meaning AND subcategorizing for either clauses or verbal phrases were 1,746 and 1,619 triggers by Annotators 1 and 2, respectively. Triggers labeled as not conveying a modality meaning and/or not modifying a clause or a verbal phrase were 1,146 and 1,273 triggers by Annotators 1 and 2, respectively. Exact matches between the two annotators (i.e. triggers labeled similarly for modality meanings, scope type(s) and scope span(s)) amount to 1,343 valid modality triggers according to our definition.

	Modal	Non-Modal
<b>Annotator 1</b>	1,746	1,146
<b>Annotator 2</b>	1,619	1,273
<b>Agreed-Upon (Exact-Match)</b>	1,343	1,034
<b>Total Exact matches</b>	<b>2,377</b>	

Table 7: Exact-match modality annotated corpus

## 5.2 Annotated Corpus Statistics

In this section, we give statistics on candidate triggers labeled identically by both annotators: whether triggers eventually considered as valid modality triggers (i.e. 1,343 triggers) or triggers eventually rejected as invalid modality triggers (i.e. 1,034). Table 8 shows the correlation between modality semantic meanings and their scope types. We conclude that:

- Except for evidential sensory, modality triggers are more likely to modify clauses and verbal phrases than other linguistic constituents such as noun, adjective and adverb phrases. That is, modality triggers subcategorize for clauses and verbal phrases.
- Propositional modality (i.e. epistemic, evidential reported and sensory) subcategorizes more frequently for clauses; whereas event modality (i.e. deontic and dynamic) is more likely to subcategorize for verbal phrases.
- Triggers that were pre-highlighted as candidates by AML and later rejected for being invalid according to our definition correlate more frequently with linguistic constituents other than clauses and verbal phrases.
- Only 2% of scopes are implicit.

Based on AML dialect labels, valid modality triggers are: (1) 77.3% EA-exclusive such as *mthy>ly* (I think) and *EAYz* (I want), (2) 15.6% either MSA or EA based on context such as *qAl* (he said) and *Erf* (he knew),

and (3) 7% MSA-exclusive such as *Astwjib* (it necessitated) and *wddt* (I wanted).

	Clause	V. Phrase	Implicit	NA	Total
<b>Ep</b>	477	23	15	33	<b>548</b>
<b>Rp</b>	197	6	7	33	<b>243</b>
<b>Sn</b>	49	1	2	120	<b>172</b>
<b>Ob</b>	54	77	15	21	<b>167</b>
<b>Pr</b>	15	47	5	11	<b>78</b>
<b>Cm</b>	2	0	0	4	<b>6</b>
<b>Ab</b>	3	97	4	9	<b>113</b>
<b>VI</b>	80	167	0	1	<b>248</b>
<b>NA</b>	36	6	0	760	<b>802</b>
<b>Total</b>	<b>913</b>	<b>424</b>	<b>48</b>	<b>992</b>	<b>2377</b>

Table 8: Modality semantic meanings and scope types

Ambiguity accounts for 37% and 5% of the valid modality lemmas for lexical and semantic ambiguity, respectively. Some of the most frequent lemmas from each ambiguity type are illustrated in Tables 9 and 10.

Lemma	Trans.	Modal Freq.	Non-Modal Freq.
<i>qAl</i>		171	40
<i>nfs</i>		56	186
<i>\$Af</i>		80	129
<i>fhm</i>	فهم	88	45

Table 9: Top frequent lexically-ambiguous lemmas

Lemma	Trans.	Modality meanings	Freq.
<i>\$Af</i>		epistemic (think)	47
		sensory (watch; witness)	95
<i>Erf</i>		epistemic (know)	41
		abilitive (can)	40
<i>mmkn</i>		epistemic (possible that)	28
		abilitive (can)	17

Table 10: Top frequent semantically-ambiguous lemmas

Finally, about 82% of the scope heads are adjacent to their triggers. This is expected given that tweets are typically short.

## 6 Conclusion and Outlook

We presented a novel modality annotation scheme and applied it to the Arabic language in the tweets genre. This work is part of a larger project to use linguistic modality to detect power relations among participants on Twitter. The presented scheme uses both semantics and syntax to increase annotation reliability. Results show that Arabic modality triggers have regular subcategorization patterns that yield high annotation agreement when used as guidelines.

Currently, we are working on an updated version of this corpus with improved guidelines to tackle disagreement factors that emerged here. The new version will also include annotations for modality holders and trigger-related features such as negation, modification and mood.

## Acknowledgement

This work has been partially supported by a grant on social media and mobile computing from the Beckman Institute for Advanced Science and Technology.

## References

- Muhammad Abdul-Mageed and Mona Diab. 2012. AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis. *The 8th International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, 21-27 May 2012.
- Rania Al-Sabbagh and Roxana Girju. 2012. YADAC: Yet another Dialectal Arabic Corpus. *The 8th International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, 21-27 May 2012.
- Kathrin Baker, Michael Bloodgood, Bonnie Dorr, Nathaniel W. Filardo, Lori Levin, and Christine Piatko. 2010. A Modality Lexicon and its Use in Automatic Tagging. *The 7th International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 19-21 May 2010.
- Virginia Brennan. 1993. *Root and Epistemic Modal Auxiliary Verbs*, PhD Thesis, University of Massachusetts, Amherst.
- Kristen E. Brustad. 2000. *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian and Kuwaiti Dialects*. Georgetown University Press, Washington DC, USA.
- John Butler. 2003. A Minimalist Treatment of Modality. *Lingua*, vol. 113 (10): 967-996.
- Joan L. Bybee, R. D. Perkins and W. Pagliuca. 1994. *The Evolution of Grammar: Tense, Aspect and Modality in the Languages of the World*. University of Chicago Press, Chicago, USA.
- Mona Diab, Lori Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran and Wei Wei Guo. 2009. Committed Belief Annotation and Tagging. *Proceedings of the Third Linguistic Annotation Workshop (ACL-IJCNLP '09)*, August 2009, Suntec, Singapore.
- Iris Hendrickx, Amália Mendes, and Silvia Mencarelli. 2010. Modality in Text: A Proposal for Corpus Annotation. *The 7th International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 19-21 May 2010
- Ray Jackendoff. 1972. *Semantic Interpretation in Generative Grammar*. Cambridge, MA: The MIT Press.
- Suguru, Matsuyoshi, Megumi Eguchi, Chitose Sao, Koji Murakami, Kentaro Inui, and Yuji Matsumoto. 2010. Annotating Event Mentions in Text with Modality, Focus, and Source Information. *The 7th International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 19-21 May 2010
- T. F. Mitchell and S. A. Al-Hassan. 1994. *Modality, Mood and Aspect in Spoken Arabic with Special Reference to Egypt and the Levant*. London and NY: Kegan Paul International.
- Ola Moshref. 2012. Corpus Study of Tense, Aspect, and Modality in Diaglossic Speech in Cairene Arabic. PhD Thesis. University of Illinois at Urbana-Champaign.
- Frank R. Palmer. 1989. *Mood and Modality*. Cambridge University Press, Cambridge, UK.
- Frank R. Palmer. 2001. *Mood and Modality*. 2<sup>nd</sup> Edition. Cambridge University Press, Cambridge, UK.
- Vinodkumar Prabhakaran. 2012. Detecting Power Relations from Written Dialog. *Proceedings of the 2012 Student Research Workshop*, Jeju, Republic of Korea.
- Vinodkumar Prabhakaran, Michael Bloodgood, Mona Diab, Bonnie Dorr, Lori Levin, Christine D. Piatko, Owen Rambow and Benjamin Van Durme. 2012. Statistical Modality Tagging from Rule-Based Annotations and Crowdsourcing. *Proceedings of the ACL-2012 Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM-2012)*, Jeju, Republic of Korea.
- Roser Saurí, March Verhagen and James Pustejovsky. 2006. Annotating and Recognizing Event Modality in Text. *The 19th International FLAIRS Conference*, pages: 335-339, Florida, USA, May 2006
- György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope Corpus: Annotation for Negation, Uncertainty and their Scope in Biomedical Texts. *BioNLP 2008: Current Trends in Biomedical Natural Language Processing*, pages: 38-45, Columbus, Ohio, USA, June 2008
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluations*, 39:165-210

# Mapping Rules for Building a Tunisian Dialect Lexicon and Generating Corpora

**Rahma Boujelbane**

ANLP Research Group, MIRACL  
Lab, University of Sfax, Tunisia  
rahma.boujelbane@gmail.com

**Meriem Ellouze Khemekhem**

ANLP Research Group, MIRACL  
Lab, University of Sfax, Tunisia  
meriem.ellouze@planet.com

**Lamia Hadrich Belguith**

ANLP Research Group, MIRACL  
Lab, University of Sfax, Tunisia  
l.belguith@fsegs.rnu.tn

## Abstract

Nowadays in Tunisia, the Arabic Tunisian Dialect (TD) has become progressively used in interviews, news and debate programs instead of Modern Standard Arabic (MSA). Thus, this gave birth to a new kind of language. Indeed, the majority of speech is no longer made in MSA but alternates between MSA and TD. This situation has important negative consequences on Automatic Speech Recognition (ASR): since the spoken dialects are not officially written and do not have a standard orthography, it is very costly to obtain adequate annotated corpora to use for training language models and building vocabulary. There are neither parallel corpora involving Tunisian dialect and MSA nor dictionaries. In this paper, we describe a method for building a bilingual dictionary using explicit knowledge about the relation between TD and MSA. We also present an automatic process for creating Tunisian Dialect (TD) corpora.

## 1 Introduction

Recently, due to the changes that have occurred in the Arab world, we noticed a new remarkable diversity in the media. The Arabic dialects used in daily life have become progressively used and represented in interviews, news and debate programs instead of Modern Standard Arabic (MSA). In Tunisia, for example, the revolution has affected not only the people but also the media.

For that reason, the media programs have been changed: television channels, political debates and broadcasts news have been multiplied. This gave birth to a new kind of language. Indeed, the majority of speech is no longer made in MSA but alternates between MSA and Tunisian Dialect (TD). Thus, we can distinguish in the same speech, MSA words, TD words and MSA-TD words such as a word with an MSA component (root) and dialectal affixes. This situation poses significant challenges to NLP: In fact, applying NLP tools designed for MSA to TD yields a significantly lower performance, making it imperative to direct research towards building resources and tools that make it possible to process this kind of language. In our case, we aim to convert this new language into text. However, this process presents a series of linguistic and computational challenges. Some of these relate to language modeling: studying large amounts of text to learn about patterns of words in a language. This task is complicated because of the total lack of TD resources, whether parallel TD-MSA text or dictionaries. In this paper, we describe a method that helps to create Tunisian Dialect (TD) text corpora and the associated lexical resources and also build a bilingual MSA-TD dictionary. This paper is organized as follows: After discussing related work, we present our method to deal with the lack of Tunisian resources (Section 3). We then proceed to discuss the method in details: we explain the manner of creating Tunisian verbal



resources (Sections 4 and 5). We present in Section 6 a tool for generating dialectal corpora. We evaluate and discuss the results in Section 7.

## 2 Related work

Arabic dialects have earned the status of living languages in linguistic studies, thus we see the emergence of a serious effort to study patterns and regularities in these linguistic varieties of Arabic (Brustad, 2000; Holes, 2004; Erwin, 1963).

To date, most of these studies have been field studies or theoretical in nature with limited annotated data. In fact, Dialectal Arabic (DA) is emerging as the language of the news and of many varieties of television programs, and also of informal communication online, in emails, blogs, discussion forums, chats, SMS, etc. In current statistical Natural Language Processing (NLP) there is an inherent need for large scale annotated resources for a language (Diab *et al.*, 2010).

But, research on computerization of DA is still in its early stages especially for TD. Several researchers have explored the idea of exploiting existing MSA rich resources to build tools for DA NLP. For example, (Chiang *et al.*, 2006) built syntactic parsers for DA trained on MSA Treebanks. Such approaches typically expect the presence of tools/resources to relate DA words to their MSA variants or translations. Given that DA and MSA do not have much parallel in terms of corpora to help translate DA-to-MSA, (Abo Bakr *et al.*, 2008) introduced a hybrid approach to transfer a sentence from Egyptian Arabic into MSA. This hybrid system consisted of a statistical system for tokenizing and tagging, and a rule-based system for constructing diacritized MSA sentences. Moreover, (Al-Sabbagh and Girju, 2010) described an approach of mining the Web to build an Egyptian-to-MSA lexicon. (Diab *et al.*, 2010) presented an information retrieval project COLABA (Cross Lingual Arabic Blog Alerts) that aims to create resources and processing tools for dialectal Arabic blogs. The COLABA system consists in taking an MSA query and translating it or its component words into DA or alternatively converting all DA documents in the search collection into MSA before searching on them with the MSA query. To do so, they created DIRA (Dialectal Information Retrieval for Arabic), which is a term expansion tool for information retrieval over dialectal Arabic collections, especially the

Egyptian and the Levantine dialects, using Modern Standard Arabic queries. (Habash and Rombow, 2006) presented MAGEAD (Morphological Analyser and Generator of Arabic dialect). MAGEAD works both on analyzing and generating Egyptian and Levantine verbs. The limitation of MAGEAD is that it doesn't deal with verbs that change their roots when moving from MSA to Dialect. (Shaalan *et al.* 2007) proposed a system for translating MSA into the Egyptian dialect. To do so, they tried to build a parallel corpus between the Egyptian dialect and MSA based on mapping rules EGY-MSA.

As a conclusion, for MSA and its dialects, there are no naturally occurring parallel corpora. It is this fact that has led researchers to investigate the use of explicit linguistic knowledge.

### Dialects are under-resourced languages:

Spoken languages which have no written form can be classified as under-resourced languages and as a consequence have no annotated resources. Therefore, several studies have attempted to overcome the problems of lack of resources for these languages. In order to computerize the existing Swiss dialect, (Scherrer, 2008) developed a translation system: standard German to Swiss German. The system developed is based on translating a bilingual lexicon from standard German to any variety of the dialect continuum of German-speaking Switzerland. Moreover, there are several languages from the group of under-resourced languages that do not have a relation with a well-resourced language. Indeed, (Nimaan *et al.* 2006) presented several scenarios to collect corpora in order to automatically process the Somali language: collecting a corpus from the Web, automatic synthesis of texts and machine translation of French into Somali. (SENG, 2010) selected news sites in Khmer to collect data in order to solicit the lack of resources in Khmer.

Related work vs. the Tunisian dialect: The literature shows that there is little work that dealt with the Tunisian dialect, the target language of this work. (Graja *et al.*, 2011) for example, treated the Tunisian dialect for understanding speech. To do so, the researchers relied on manual transcripts of conversations between agents at the train station and travelers. The scope of application is limited and so, the vocabulary is not very rich. However, a limited vocabulary is a problem if we want to model a language model for a system of recognition of

television programs with a wide and varied vocabulary. In addition, (Zribi *et al.*, 2013) presented OTTA (Orthographic Transcription for Tunisian Arabic), a set of guidelines orthography to transcribe Tunisian Arabic. This work is helpful for our case in that it will facilitate the identification of the orthography of the Tunisian words that we will build.

### 3 Method to create a Tunisian Dialect lexicon

In Arabic, there are almost no parallel corpora involving the Tunisian Dialect and MSA. Therefore, Machine Translation (MT) is not easy, especially when there are no MT resources available such as a naturally occurring parallel text or a transfer lexicon. So, to deal with this problem, we propose to leverage the large amount of annotated MSA resources available by exploiting MSA/dialect similarities and addressing known differences. Our approach consists first in studying the morphological, syntactic and lexical differences by exploiting the Penn Arabic Treebank. Second, we present these differences by developing rules and building dialectal concepts. Finally, we define a lexical data base to store these transformations into dictionaries.

#### 3.1 Tunisian Dialect Vs. MSA

The Tunisian Arabic dialect is attached to the Arab Maghreb and is spoken by twelve million people living mainly in Tunisia. It is generally known to its speakers as the 'Darija' or 'Tounsi' which simply means "Tunisian", to distinguish it from Modern Standard Arabic (Baccouche, 1994).

The Tunisian dialect is considered as an under-resourced language. It has neither a standard orthographic or written text nor dictionaries.

Actually, there is no strict separation between Modern Standard Arabic (MSA) and its dialects, but a continuum dominated by mixed forms (MSA-Dialect). In the last two years, this dialect became the language spoken in most of the media instead of standard Arabic. But this dialect has a sophisticated form which mixes MSA and TD forms. Thus, given the similarities between TD and MSA, the resources available to MSA can be advantageously used to create dialectal resources.

#### 3.2 Penn Arabic Treebank corpora to create a bilingual MSA-TD lexicon

Treebanks are important resources that allow for important research in general NLP applications. In the case of Arabic, two important treebanking efforts exist: the Penn Arabic Treebank (PATB) (Maamouri *et al.*, 2004; Maamouri *et al.*, 2009) and the Prague Arabic Dependency Treebank (PADT) (Smrř *et al.*, 2008). The PATB provides tokenization, complex POS tags, and syntactic structure; it also provides empty categories, diacritizations, and lemma choices. The ATB consists of 23,611 parse-annotated sentences (Bies and Maamouri, 2003; Maamouri and Bies, 2004) collected from Arabic newswire texts in Modern Standard Arabic (MSA). The ATB annotation scheme involves 497 different POS-tags with morphological information. In this work, we attempted to mitigate the genre differences by transforming the MSA-ATB to look like TD-ATB. This will allow creating in tandem a bilingual lexicon with different dialectal concepts (Figure1). For this purpose, we

adopted a transformation method based on the parts of speech of ATB's words, as discussed in the following.

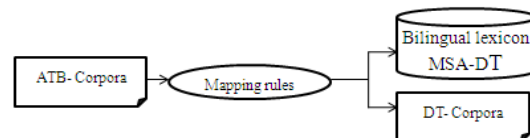


Figure1- Method for creating TD resources

#### 4 Mapping rules based on verbal morphological distinction

There's a difference between verb conjugation in MSA and that in TD. We find that in TD, the gender distinction is not marked. Most Tunisian people do not distinguish between masculine and feminine with the second person-singular. Similarly, we mark the absence of the masculine and feminine dual. Another conjugation difference is in the passive form of the TD and MSA verb. In fact, the passive form of most Tunisian verbs is obtained by preceding the verb with the consonant 'ت' [t]. Unlike in MSA, passive verbs in TD cause the transformation of the structure of the sentence: For example, the transformation of the sentence (Active voice) **كلا الطفل التفاحة**/kIA aITfol AltofeHap/The boy ate the

apple/ is in passive voice “التفاحة تاكلت”/AltofeHa teklit/The apple has been eaten/

In the imperfect, the [t] lies between the root and the prefix as in the following:

"يتاكل"/yitekil/ The lunch (M) is edible. *Masculin*

/"تتاكل"/titekil/The apple (F) is edible *Feminin*. In addition to this type of form, the dialect offers another form frequently used as question such as: "تتاكلشي"/titekil\$y/ Is it edible?

In this work, as we aim to build a lexicon for Tunisian verbs, we must take into account these differences. But to define Tunisian inflected forms, we should first define the main concept of “Arabic verb” and we will do this by studying the morphological and lexical differences that may exist between TD verbs and MSA verbs. Indeed, in Arabic there are three principal verbal concepts:

1-Root: It is the basic source of all forms of Arabic verbs. The root is not a real word; rather, it is a sequence of three consonants that can be found in all words that are related to it. Most roots are composed of three letters; very few are composed of four or five consonants.

2-Pattern: In MSA, patterns are models with different structures that are applied to the root to create a lemma. For example, for the root خ ر ج : x r j, we can apply different patterns which give different lemmas with different meanings:

Root1: x r j /خ ر ج / C1C2C3+ verbal pattern1: AistaC1oC2a3 =lemma1 اسْتَخْرَجَ /to extract

Root1: x r j /خ ر ج / C1C2C3+ verbal pattern2 FoEaL (FaEal)=lemma2 خَرَجَ /to go out .

Root1: x r j (خ ر ج) / C1C2C3+ verbal pattern3 >aC1oC2aC3=lemma3 أَخْرَجَ /to eject

3-Lemma: The lemma is a fundamental concept in the processing of texts in at least some languages. An Arabic word can be analyzed as a root inserted into a pattern.

#### 4.1 Verbal concepts for the Tunisian dialect

As we aim to adapt MSA tools to TD, we tried to build for TD verbs the same concepts as those in MSA. Therefore, we focused in this work on the study of correspondences that may exist among the concepts of MSA verbs and dialect verbs. First, we extracted all the verbs that exist in ATB, represented in their inflected forms. Second, we used a lemmatizer to extract lemmas; we obtained as a result 1500 different MSA lemmas. Third, we built manually lemmas

corresponding to TD. Later, we tried to build verbal patterns equivalent to those in MSA. Finally, since there is no standard definition of roots in TD, we opted for a deductive method to define root for dialect verbs. Figure 2 illustrates this method.

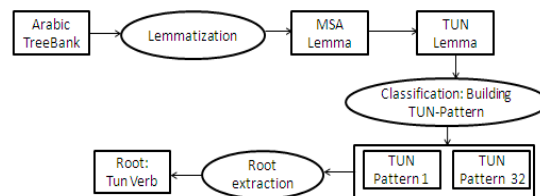


Figure 2: From ATB verb to TD-verb

**Building TD-lemmas:** Verbs in the ATB corpus are presented in their inflected forms. So, we extracted lemmas and their roots using the morphological analyzer developed by Elixir FM (Smrz, 2007). As we are native speakers of TD, we associated to each MSA-Lemma a TUN-Lemma. As a result, we found that 60% of the verbs change totally when passing from MSA to TD. This is a preliminary step for building Tunisian patterns from which we will be able to deduct the inflectional forms. So, as we have 1500 TD-Lemmas, and starting from the fact that MSA verbs have patterns describing their morphological behavior during conjugation, we tried, whenever possible, to define to each TD-Lemma a TD-Pattern which is similar to the MSA-pattern.

**Building TD-patterns:** The challenge in building TD-patterns was to find patterns similar to those in MSA. In MSA, patterns are models with different structures that are applied to the root to create a lemma. In fact, for trilateral roots there are in MSA ten patterns I: CCC, II: CaC~aC, III: CACaC, IV: >aCCaC, V: /taCaC~aC, VI: taCACaC, VII: AinCaCaC, VIII: AiCtaCaC, IX : AiCCaC~, X: AistaCCaC.

To classify the lemmas that we have already built, we focused on the creation of verbal patterns for TD verbs. So, we chose three criteria that classify verbs from general (without considering the vowels of the word) to specific (dealing with the different variations of vowels in its conjugation).

#### Classification according to the verb model

Verb model means the form that the root takes after applying the Pattern, for example:

Root :خ ر ج /x r j ; Pattern: CaCaC; Lemma : خَرَجَ /xaraj ; Model : CVCVC

Root: خ ر ج /xrj ; Pattern: AistaCCaC ; Lemma: اسْتَخْرَجَ/Aistaxraj ; Model : AistaCVCVC  
 Classification according to the model of the verb consists in studying similarities between verb models without considering changes in vowels. Indeed, as we have already mentioned, we have 40% of verbs that do not change their root when the pass from MSA to TD. They therefore have the same model without considering vowels. To do this, we assigned to TD-verbs patterns equivalent to those in MSA (1).

For example: MSA-lemma: خَرَجَ / xaraj/go out  
 Pattern-MSA: CaCaC Model: CVCVC

→ TD: lemma: xoraj Model: CVCVC then  
 Pattern-TUN: CoCaC

Moreover, for verbs that change their root when passing to the dialect, we reasoned as follows: For a TD verb whose model looks like the model of a TD-verb for which we have already assigned a Tun-pattern (1), we assign the same Tun-pattern (2).

Example1:

MSA: صَمَتَ /Samat /be silent → TD: سَكَّتَ /sokut  
 Model : CVCVC looks like the model of خَرَجَ / xraj/: go out : CVCVC. (1)

We have already assigned to خَرَجَ / xoraj the -TUN-pattern: CoCaC. Therefore, سَكَّتَ / sokut will have the pattern -TUN: CoCuC (2).

In this way, we classified almost all TD verbs except a few who have a complex form illustrated by a verbal unit plus another lexical unit (particle or other...).

For example, the translation of the MSA verb رَافَقَ /rAfaqa/go with → is in TD: مَشَى-مع /mo\$ay-moEa. We associated this type of verb to patterns that we called "exception patterns"

### **Classification according to the vowel of the second consonant of the pattern**

The vowel of the second consonant of the pattern (vowel letter ع / E) is a fundamental criterion for classifying a verb in MSA (Ouerhani, 2009). In fact, according to this criterion, the MSA pattern I is divided into six patterns due to the variation of the vowel of the second consonant (both in past and present tense). These patterns are respectively: I-au: CaCa-yaCoCuC ; I-ai : CaCaC-yaCiC ; I-aa: CaCaC-yaCoCaC, I-ia: CaCiC-yaCoCaC ; I-uu: CaCuC-yaCoCuC ; I-ii: CaCiC-yaCaCiC.

In TD, this variation is very common and it is marked not only in the pattern I but in all patterns. For this reason, we proposed to divide these patterns and to define new patterns in order

to consolidate the verbs that have the same behavior. For example, for the Pattern-TUN II: MSA: Pattern-TUN II: no TD sub-pattern: New three sub-patterns: II-aa: CaC~aC/yiCaC~aC ; II-ai: CaC~aC /yiCaC~iC ; TUN II-ii: CaC~iC /yiCaC~iC

### **Classification according to the Imperfect mark**

The third classification criterion is based on the imperfect mark. In MSA, this mark remains unchanged in all verbs belonging to the same class. In fact, for the MSA pattern I CaCaL/yaCCAC, the mark is ي/ya ; for example: كَتَبَ /kataba-yaktubu/write. For the pattern III/CACaC/yaCACiC, the mark is يُ/yu; for example يَشَارِكُ-شَارَكَ \$Araka-yu\$Ariqu/participate.

However, we noticed that in TD, this regularity appears especially in the pattern I, so this mark can vary even within the same class. For example, يَخْرُجُ - خَرَجَ / xraj-yuxruj/to go out belongs to theTUN –pattern-I-au; يَقُولُ / قال - QAI-yiquwl/to say belongs to the TUN-pattern-I-au. Note that although these two verbs belong to the same class, their imperfect marks are different. For this reason, we proposed to extend the TUN-pattern-I-au and define more sub-patterns for the pattern I.

In this way, we assigned to يَخْرُجُ - خَرَجَ / xraj-yuxruj the pattern- I -au-u and to يَقُولُ / قال - QAI-yiquwl the pattern- I -au-i.

The result of this classification has allowed distinguishing 32 patterns for dialect verbs while there were 15 in MSA.

### **-TD-root definition**

In Tunisian dialect, there is no standard definition for the root. For this reason, dialect root construction was not obvious, especially when the verb root changes completely from the MSA to the dialect. In fact, to define a root for TUN verbs, we adopted a deductive method. Indeed, in MSA, the rule says: root + pattern =Lemma (1). In our case, we have already defined the TUN-lemma and the Tun-pattern. Following rule (1), the extraction of the root is then made easy. For example, we classified the lemma اِسْتَيْ / Aistan ~ aY / Wait in the pattern AistaCCaC then root(?) + AistaCCaC= Aistan~ Y

Following (1), the root is "نني" [NNY]. In fact, we can say that the definition of roots is a problematic issue which could allow more discussion. According to (1), it was as if we had forced the roots to be [NNY]. However, if we

classify إستنى / Aistann ~ aY under the pattern AiCtaCaC, the root in this case must be سنن / snn. The root can also be quadrilateral سنني / snnY if we classify Aistann ~ aY under the pattern AiCCaCaC. But as there's no standard, we did our best to be as logical as possible to define dialectal root.

## 4.2 Verbal lexicon structure

The various verbal transformations described above are modeled and stored in a dictionary of verbs as follows: to each MSA verbal block containing the MSA-lemma, the MSA-pattern and the MSA-root will correspond a TD- block which contains the TD-lemma, the TD-root- and the TD-pattern. So, knowing the pattern and the root, we will be able to generate automatically various inflected forms of the TUN verbs. That's why we also stored in our dictionary the active and the passive form of the TD-lemma in perfective and imperfective tenses. We also stored the inflected forms in the imperative (CV). Figure 3 shows the structure that we have defined for the dictionary to present the TD-verbal concepts (in Section 4, we will explain how we will automate the enrichment of this dictionary).

```
<DIC_TUN_VERBS_FORM>
<LEXICAL-ENTRY POS="VERB">
<VERB ID-VERB="48">
  <MSA-LEMMA>
    <Headword-sa>عَايَنَ</Headword-MSA
    <Pattern>فاعل</Pattern>
    <Root-Msa>عين</Root-Msa>
    <Gloss lang="ang" > Observe</Gloss>
  </MSA-LEMMA>
  <TUN-VERB Sense= "1" >
    <Cat-Tun-Verb Category=
      TUN-VERB--I-au--yi" />
  <Root-Tun-Verb>شوف</Root-Tun-Verb>
  <Conjug-Tun-Verb>
  <TENSE>
  <FORM Type= "IV" >
  <VOICE Label="Active">
  <Features Val_Number_Gender="1S">
  <Verb_Conj>نشوف</Verb_Conj>
  <Struct-Deriv>∅+شوف+ن</Struct-Deriv>
  </Features>
  </VOICE>
  ...
</DIC_TUN_VERBS_FORM>
```

Figure3- Verbal dictionary structure

## 5 Mapping rules based on syntactic distinction

We identified three areas that reflect the specific syntax of the dialect: word order, grammatical negation and syntactic tools categories. In the following section, we will explain how we define these dialect structures in our lexicon.

### 5.1 Word order

The order of the elements in the dialect sentence seems to be relatively less important than in other languages . However, the canonical word order in Tunisian verbal sentences is SVO (Subject-Verb-Object) (Baccouche, 2003).

In contrast, the MSA word order can have the following three forms: SVO / VSO / VOS (2).

(1) TD: « الطُّفْلُ كَتَبَ الدَّرْسَ ».SVO

(2) MSA: « كتب الطفل درس ».VSO.

This opposition between MSA and the dialect is clearer in the case of proper names. In fact, MSA order is VSO (3) while the order in TD is SVO. (Mahfoudhi, 2002)

(3) MSA : « ضرب موسى عيسى ».

(4) TD : « موسى ضَرَبَ عيسى ».

There are other types of simple dialect sentences named nominal sentences which do not contain a verb. They have the same order in both TD and MSA. For example:

MSA: الطقس حار /TaKs HAR/ The weather is hot

TD: الطَّقْسُ سَخُونُ /TaKs sxuwn/The weather is hot

In our work, we discussed some nominal groups at the syntactic level. The word order is generally reversed when passing to TD. For example

(1) MSA: ADV + ADJ

>ayDaA/Also+مُنَقَّف /muvaK~af/also educated

TD: ADJ +ADV

ADJ/ مُنَقَّف +ADV/زاده

(2) MSA: Noun + ADJ

MSA: كُتُبٌ كَثِيرَةٌ /kutubun kavira/many books->

TD: ADJ + Noun

TD: برشا كُتُب /bar\$A ktub

In the dictionary, we present this kind of rule as shown in Figure 4.

```
<ADV-MSA ID="5">
  <MSA-LEMMA>أَيْضاً</MSA- LEMMA>
  <GLOSS ang="ang">Also</GLOSS>
  <CONTEXT ID="1">
    <CONFIG ID="1" Position="Before" POS="ADJ" />
    <TOKEN>
      <TUN ID="1" DIC="ADJECTIVES" POS="ADJ" />
      <TUN ID="2" />
      <TUN ID="3">زَادَا</TUN>
    </TOKEN>
```

</CONTEXT>

Figure 4- Syntactic rule representation in the dictionary

## 5.2 Grammatical negation

Negation particles are generally set before the verb and can sometimes change the combination. For example, if the word <أكتب> />ktb /Write in MSA is preceded by a negative particle such as لم/lam (Do not), the verb in the dialect will be: mAktibtibti\$/ماكتبتيش= Tun-Neg-Particle(ما)+ Tun-verb (كتب)+ Tun-Neg-enc (ش)

## 5.3 Syntactic tool categories

Tools words or Syntactic tools exist in a large amount in the Treebank and all MSA-texts. However, their transformation was not trivial and required for each tool a study of its different contexts.

A tool word may have different translations depending on its context. For example, the particle حَتَّى/ HatY/so that: we found this particle in ATB in three contexts. This particle gives a new translation whenever it changes context:

- 1- حَتَّى/ HatY + verb = باش (TUN-particle) + TUN\_verb
- 2 حَتَّى/ HatY + NEG\_PART = باش (TUN-particle) + TUN\_NEG\_PART
- 3- حَتَّى/ HatY = حَتَّى/ HatY

So, to deal with these transformations, we converted them into rules and stored them into a lexicon of tool word transformation.

### Context dependent transformation

We mean by context dependent transformation the passage MSA-TD which is based on transformation rules. Indeed, given the word MK, we say that the transformation of MK is based on context if it gives a new translation whenever it changes context. RTK : X + M + Y = TDk

$$X = \sum_{j=1}^m Mj: POSj ; Y = \sum_{i=1}^n Mi: POSi ; k \text{ varies from } 1 \text{ to } z ;$$

RTk: transformation rules n°k; POS : Part of speech ; M: word tool, TDk: Translation n°k

The transformation of a tool word may depend on the words (X) that precede it, or on the following word (Y), or both. If none of the

contexts is presented, then a default translation will be assigned to the tool word. In total, we defined in the tool words dictionary 316 rules for the 146 ATB's tool words.

In the dictionary, we presented a transformation rule. In fact, for each tool word we defined a set of contexts; each context contains one or more configurations. The configuration describes the position and the part of speech of the words of context. Each context corresponds to a new translation of the tool word (Figure 5).

```
<PREP-MSA ID="9">
  <MSA-LEMMA>حَتَّى</MSA-LEMMA>
  <GLOSS lang="ANG ">until </GLOSS>
  <CONTEXT ID="1">
    <CONFIG ID=" 1 " Position="Before" PRC="DET" />
      <CONFIG ID="2" Position=" Before "
        POS="NOUN">ساعة</CONFIG>
  <CONFIG ID="3" Position=" Before" POS="NOUN_NUM" />
  <TOKEN>
    <TUN ID="1">حَتَّى</TUN>
    <TUN ID="2" POS="NOUN_NUM" />
  </TOKEN>
</CONTEXT>
.....
<CONTEXT ID="6">
.....
</Prep-MSA>
```

Figure5- Structure of a context dependent rule in the dictionary

### Context independent transformation

In addition to the context-dependent transformations, the translation of some tool words in the corpus was direct "word to word"; the word remained the same regardless of the context. Figure 6 shows an example of how we represented this kind of translation in the dictionary

```
<SUB_CONJ-MSA ID="7">
  <MSA-LEMMA>كَي</MSA-LEMMA>
  <GLOSS lang="ANG">In order to
</GLOSS>
<TOKEN>
  <TUN ID="1">بَاشْ</TUN>
</TOKEN>
</SUB_CONJ-MSA>
```

Figure 6- Structure of a context independent rule in the dictionary

## 6 Automatic generation of Tunisian Dialect corpora

To test and improve the developed bilingual models, we exploited our dictionaries to



automate the task of converting MSA corpora to corpora with a dialect appearance.

For this purpose, we developed a tool called Tunisian Dialect Translator (TDT) which enables to produce TD texts and to enrich the MSA-TD dictionary (Figure 6). The TDT tool works according to the following steps:

1-Morphosyntactic annotation of MSA texts: TDT annotates each MSA text morphosyntactically by using the MADA analyzer (Morphological Analyser and disambiguator of Arabic) (Habash, 2010). MADA is a toolkit which, given a raw MSA text, adds lexical and morphological information. It disambiguates in one operation part-of-speech tags, lexemes, diacritizations and full morphological analyses.

2-Exploiting MSA-TD dictionaries: Based on each part of speech of the MSA-word, TDT proposes for each MSA structure the corresponding TD translation by exploiting the MSA-TD dictionaries.

3-Enriching the lexicon: As our MSA-TD dictionaries do not cover all Arabic words, texts resulting from the previous step are not totally translated. Therefore, in order to improve the quality of translation and to enrich our dictionaries, enabling them to be well used even in other NLP applications, we added to TDT a semi-automatic enrichment module. This module filters first all MSA words for which a translation has not been provided. Then, TDT assigns to them their corresponding MSA-lemmas and POS. If the POS is a verb or a noun, the user proposes a TD-root and a TD-pattern (described in subsection 3.2) and the TDT generates automatically the appropriate Tunisian lemma and its inflected forms.

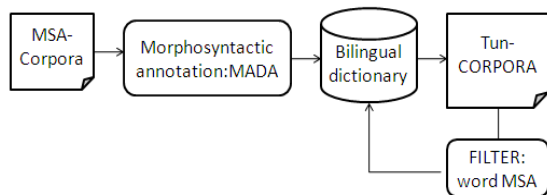


Figure7- Automatic process for generating Tunisian corpora

## 7 Evaluation

To evaluate translations of verbs occurring in our MSA-TD dictionary, we asked 47 judges (native speakers) to translate for us a sample containing 10% of verbs extracted from the dictionary. In this sample, there are 52 verbs that don't change their root when passing to TD and 98 do otherwise. The evaluation consists in

comparing what we have proposed as a translation of lexical items taken from the ATB with the proposals of judges who are native speakers of the Tunisian dialect. The percentage of agreement between the judges' translations and the translations proposed in our lexicon is calculated. Table 1 shows the results obtained

Verbs	Unchanged	Changed	Total
Number of verbs in the sample	52	98	150
Agreement	97,17%	63,21%	74,97%

Table 1- Evaluation of verb translation

Moreover, as the translation of the majority of tool words depends on context, we asked 5 judges to translate 89 sentences containing 133 tool words. In this sample, we repeated some tool words in the same sentence but in a different context. Table (2) gives the percentages of agreement between the translations of the judges and those in our dictionaries of tools words. The variation in percentage is due to the fact that for some words, the judges do not agree among themselves. The table shows also the percentage of disagreement between the judges and the dictionaries.

	2 judges	3 judges	4 judges	5 judges
Agreement	72,69 %	74,53 %	71,34 %	71,23 %
<b>Disagreement</b>	18,79 %	15,03 %	14,28 %	12,03 %

Table 2- Evaluation of tool word translation

In fact, disagreement arises when no judge gives a translation similar to the translation proposed in the dictionaries. But, by increasing the number of judges, the disagreement decreases, which proves that our dictionaries contain translations accepted by several judges

## 8 Conclusion

This paper presented an effort to create resources and translation tools for the Tunisian dialect. To deal with the total lack of written resources in the Tunisian dialect, we described first a method that allowed the creation of bilingual dictionaries with in tandem TD-ATB. In fact, TD-ATB will serve as a source of insight on the phenomena that need to be addressed and as corpora to train TD-NLP tools. The verb dictionaries and the verbal concepts that we have developed were also exploited in order to adapt MAGEAD

(Habash *et al.* 2006) (Morphological Analyser and Generator of Arabic Dialect) to the Tunisian dialect (Hamdi *et al.*, 2013).

We focused second on describing TDT, a tool used to generate automatically TD corpora and to enrich semi-automatically the dictionaries we have built.

We plan to continue working on improving the TD-resources by studying the transformation of nouns. We also plan to validate our approach by measuring the ability of a language model, built on a corpus translated by our TDT tool, to model transcriptions of Tunisian broadcast news.

Experiments in progress show that the integration of translated data improves lexical coverage and the perplexity of language models significantly.

## References

- Al-Sabbagh Rania and Girju Roxana. 2010. *Mining theWeb for the Induction of a Dialectal Arabic Lexicon*. In Nicoletta Calzolari.
- Bies Ann. 2002. *Developing an Arabic Treebank: Methods , Guidelines , Procedures , and Tools*.
- Baccouche Tayeb. 1994. *L'emprunt en arabe moderne*, Beit Elhikma et IBLV, Tunis.
- Baccouche Tayeb. 2003. *La langue arabe: Spécificités et évolution*.
- Brustad Kristen. 2000. *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects*. Georgetown University Press.
- Chiang David, Diab Mona, Habash Nizar, Rambow Owen and Shareef Safiullah. 2006. *Parsing Arabic Dialects*. In Proceedings of the European Association for Computational Linguistics (EACL).
- Chiang David, Diab Mona, Habash Nizar, Rambow Owen and Safiullah Shareef. 2006. *Parsing Arabic Dialects*. In Proceedings of the European Chapter of ACL. EACL.
- Diab Mona, Habash Nizar, Owen Rambow, Al Tantawy Mohamed and Benajiba Yassine. 2010. *COLABA: Arabic Dialect Annotation and Processing*. LREC Workshop on Semitic Language Processing, Malta, May 2010.
- Graja Marwa, Jaoua Maher and Belguith Lamia. 2011. *Building ontologies to understand spoken*, CoRR.
- Habash Nizar, Rambow Owen and Roth Ryan. 2009. *MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization*. In Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt.
- Habash Nizar and Rambow Owen. 2005. *Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop*. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics ACL'05, pages 573–580, Ann Arbor, Michigan.
- Habash Nizar and Rambow Owen. 2006. *Magead: A morphological analyzer for Arabic and its dialects*. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling-ACL'06), Sydney, Australia.
- Hamdi Ahmed, Boujelbane Rahma, Habash Nizar and Nasr Nizar. 2013. *Un système de traduction de verbes entre arabe standard et arabe dialectal par analyse morphologique profonde*. TALN, Nante, France.
- Hitham Abo Bakr, Shaalan Khaled and Ibrahim Ziedan. 2008. *A hybrid approach for converting written egyptian colloquial dialect into diacritized Arabic*. In the 6th International Conference on Informatics and Systems, INFOS. Cairo University.
- Holes Clive. 2004. *Modern Arabic: Structures, Functions, and Varieties. Georgetown Classics in Arabic Language and Linguistics*. Georgetown University Press.
- Marcel Diki-kidiri. 2007. *Comment assurer la présence d 'une langue dans le cyberspace*.
- Maamouri Mahmoud and Bies Ann. 2004. *Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools*, Workshop on Computational Approaches to Arabic Script-based Languages, COLING.
- Maamouri Mahmoud, Bies Ann, Krouna Sondes, Kulick Seth, Mekki Wigdan and Buckwalter Tim. 2009. *Penn arabic treebank guidelines with much appreciated contributions from*, 1–248.
- Maamouri Mohamed, Bies Ann, Kulick Seth, Zaghouani Wajdi, Graff David and Ciul Michael. 2010. *From Speech to Trees: Applying Treebank Annotation to Arabic Broadcast News*, Lrec.
- Mohamed Emad, Mohit Behrang and Oflazer Kemal. 2012. *Transforming Standard Arabic to Colloquial Arabic*, (July), 176–180.
- Mahfoudh Abdessatar. 2002. *Agreement lost Agreement Regained: A Minimalist Account of Word Order and Agreement Variation in Arabic*, University of Ottawa.
- Nimaan Abdillahi, Nocera Pascal and Orres-Moreno Juan-Manuel. 2006. *Boîte à outils TAL pour des*



*langues peu informatisées: le cas du Somali*, JADT.

Ouerhani Bechir, *Interférence entre le dialectal et le littéral en Tunisie: Le cas de la morphologie verbale*, 75–84.

Scherrer Yves. 2008. *Transducteurs à fenêtre glissante pour l'induction lexicale*, Genève

Seng Sopheap, Sam Sethserey, Le Viet-Bac, Bigi Brigitte and Besacier Laurent. 2010. *Reconnaissance automatique de la parole en langue khmère : quelles unités pour la modélisation du langage et la modélisation acoustique*.

Smrž Otakar. 2007. *Computational Approaches to Semitic Languages*, ACL, Prague

Smrž Otakar, Viktor Bielický, Iveta Kourilová, Jakub Kráčmar, Jan Hajic and Petr Zemanek. 2008. *Prague Arabic Dependency Treebank: A Word on the Million Words*.

Zribi Ines, Graja Marwa, Ellouze Khmekhem Mariem, Jaoua Maher and Hadrich Belguith Lamia. 2013. *Orthographic Transcription for Spoken Tunisian Arabic*, CICLing, Samos, Greece.

# Hypothesis Refinement Using Agreement Constraints in Machine Translation

**Ankur Gandhe**

Carnegie Mellon University,  
USA

ankurgan@andrew.cmu.edu

**Rashmi Gangadharaiah**

IBM Research,  
India

rashgang@in.ibm.com

## Abstract

Phrase-based machine translation like other data driven approaches, are often plagued by irregularities in the translations of words in morphologically rich languages. The phrase-pairs and the language models are unable to capture the long range dependencies which decide the inflection. This paper makes the first attempt at learning constraints between the language-pair where, the target language lacks rich linguistic resources, by automatically learning classifiers that prevent implausible phrases from being part of decoding and at the same time adds consistent phrases. The paper also shows that this approach improves translation quality on the English-Hindi language pair.

## 1 Introduction

Data driven Machine Translation approaches have gained significant attention as they do not require rich linguistic resources such as, parsers or manually built dictionaries. However, their performance largely depends on the amount of training data available (Koehn, 2005).

When the source language is morphologically rich and when the amount of data available is limited, the number out-of-vocabulary (OOV) increases thereby reducing the translation quality. Popovic and Ney (2004) applied transformations to OOV verbs. Yang and Kirchoff (2006) used a back-off model to transform unknown words, where, the phrase-table entries were modified such that words sharing the same root were replaced by their stems. Others (Freeman et al., 2006; Habash, 2008) found in-vocabulary words that could be treated as morphological variants.

Translating into a language that is rich in morphology from a source language that is not morphologically rich also has limitations. The main

reason for this is that the source language does not usually contain all the information for inflecting the words in the target half. For language-pairs that have limited amounts of training data, it is unlikely that the Translation model comes across all forms of inflections on the target phrases. Hence, some mechanism is required in order to generate these target phrases with all possible inflections and at the same time be able to filter out the implausible hypotheses.

Certain approaches (Toutanova et al., 2008; Minkov et al., 2007; Green et al., 2012) predict inflections using syntactic and rich morphological sources for the target language. This approach cannot be applied on resource poor languages such as, Hindi or other Indian languages, which lack such rich knowledge sources. Ramanathan et al. (2009) use factored models to incorporate semantic relations and suffixes to generate inflections and case markers while translating from English to Hindi but do not consider the problem of agreement between phrases in the target sentence. William and Koehn (2011) suggested an approach to eliminate inconsistent hypotheses in a string-to-tree model by adding unification-based constraints to only the target-side of the synchronous grammar. Although transfer-based MT (Lavie, 2008) uses rich feature structures, grammar rules and constraints are manually developed. In addition, rules formed for one language-pair cannot be applied to another language pair. However, it is possible to model these rules as a classification problem: Given the set of source language features that influence the inflection of the target word, we try to predict the best possible target class. The target class could be the either spontaneous words or inflections of words.

This paper, specifically looks at translating from English to Hindi to predict a) Subject case markers, b) Object case markers and c) Verb phrase inflections. In many PBSMT systems, once the

phrase-pairs have been extracted, it is no longer required to store the training corpus from which the phrase-pairs were extracted. However, while dealing with many morphologically rich languages, the morphological variants of the target phrase not only depend on their source phrase but also on the context in which the source phrase appeared. Hence, it is beneficial to incorporate source-side features while decoding and most PBSMT systems do not use any other information from the input sentence other than the source phrase itself.

This paper presents an approach to improve the translation quality while translating from a morphologically poor language (such as, English) to a target language that is morphologically rich without using any rich resources such as, parsers or morphological analyzers. The contributions of the paper are summarized as follows:

- The approach detects inconsistent hypotheses generated by the translation model by treating the task as a classification problem.
- The approach also predicts plausible target phrases that agree with the features extracted from the input sentence.
- The paper also shows how the incorporation of source-specific features during decoding results in better translations.

Section 2 provides motivating examples to understand the importance of the task at hand.

## 2 Motivation

We demonstrate the usefulness of our approach on Indian languages as they are rich in morphology. They are also considered as resource-poor and low-density languages due to the lack of data availability and the absence of rich knowledge sources like morphological analyzers or syntactic parsers. Hindi has a free word-order where the constituents are identified through case markers.

A few approaches generate the right inflection by a) capturing all possible variations within the target phrase (Gandhe et al., 2011) and b) use the language model to select the most fluent phrases. However, the following problems still remain:

1) Many language models typically use 4-gram or 5-gram models (even lower when the data available is scarce). Example 1a has a subject (Ram) that is masculine (masc)-3rd person (3)-singular(sg)-present progressive(pp) and example

1b, has a subject (Sita) that is feminine (fem)-3rd person(3)-singular(sg)-present progressive(pp). This difference in gender, changes the inflection on the auxiliary Hindi verb *raha*, from ‘a’ (in 1a) to ‘i’ (in 1b). It should be noted that lower order n-gram language models fail to obtain the right translation due to the long distance dependency between the subject (*Ram / Sita*) and the verb phrase (*khel raha hai / khel rahi hai* corresponding to *is playing* in English) in the target language.

### Example 1a:

S: Ram is playing with the grand master .  
 T: *Ram grand master ke saath khel raha hai* .  
 (Ram grand master with play+3+sg+masc+pp)

### Example 1b:

S: Sita is playing with the grand master .  
 T: *Sita grand master ke saath khel rahi hai* .  
 (Sita grand master with play+3+sg+fem+pp)

2) Language models are insufficient to produce the right inflections. Consider the case shown in example 2, where the translation of the English pronouns (*he/she*) is same in Hindi (both translate to *Woh*). The inflection on the auxiliary verb phrase (*raha hai / rahi hai*) is still being decided by the gender of the subject (*he/she*). Even if a higher order language model is employed, the language model gives equal preference to both the translations as the information about the gender of the subject is completely absent in the Hindi translation. Hence, the information that *Woh* corresponds to masculine in example 2a and feminine in example 2b has to come from the source sentence (*He/She*).

### Example 2a:

S: He is playing chess .  
 T: *Woh chess khel raha hai* .  
 (he chess play+3+sg+masc+pp)

### Example 2b:

S: She is playing chess .  
 T: *Woh chess khel rahi hai* .  
 (she chess play+3+sg+fem+pp)

3) Most often in PBSMT systems, the subject and verb phrases are far apart and hence are extracted independently, as in the case of example 1. Since there are no constraints during decoding on which phrases to choose, mis-matched phrases

may get picked. Apart from verb inflections, the presence of the case-marker ‘*ne*’ (shown in example 3) on the subject blocks the transfer of the subject’s gender onto the verb phrase and the verb phrase instead gets inflected with the gender of the object(*apple*). This blocking/presence of case markers is also not captured by traditional PBSMT systems.

#### Example 3a:

S: He ate an apple .

T: *us ne seb khaya* .  
(he apple ate+3+sg+masc+past)

#### Example 3b:

S: She ate an apple .

T: *us ne seb khaya* .  
(she apple ate+3+sg+masc+past)

### 3 Model

The agreement constraints can be applied to either the translation model or the language model, such that implausible combination of phrases are not picked for the best hypothesis. In our approach, we apply the agreement constraints on the translation model by filtering phrase-pairs which have an incorrect inflection on the target phrase. Since the problem of inconsistent output is mainly due to the subject, object and verb phrases, we determine agreement constraints only for these target words. For instance, suppose a ‘female’ gender inflection is expected on the target verb. Then, any phrase that contains ‘male’ gender inflection on the verb will produce an inconsistent translation and hence should be penalized. We can also add phrase-pairs when the correct inflection is not present in the phrase table.

The easiest way to filter the inconsistent phrase-pairs is to create manual rules to look at the English source side that specify the possible set of target translations and discard the rest. For instance, using example 3 in Section 2, we could create a manual rule, “When the English verb tense is ‘*past*’, Hindi subject takes the case marker ‘*ne*’ and the verb phrase takes the gender and number of the ‘subject’ ”. However, this is time consuming and it is difficult to create an exhaustive list of such rules. Hence, it is imperative that we learn these rules from data. In this paper, we use multi-class support vector machine (Crammer and Singer, 2001) classifiers that use features only

from the input source sentence to predict possible target case marker/inflections for the subject, object and verb phrases in the target sentence. We treat these as the allowed inflections on the target phrases and penalize phrase-pairs that do not contain the predicted target inflections. This methodology is expected to prevent implausible sentences being translated and improve the overall fluency of the translated sentence.

## 4 Classification

We model the prediction of the possible target inflections for a given input sentence as a classification problem. We build different classifiers<sup>1</sup> to predict the target inflections of parts of the input sentence for which the translations are dependent on long range morphological rules. The features that we use for the different classifiers are listed in Section 5. The classifiers built are as follows:

**Subject Classifier (SubCM) and Object Classifier (ObjCM):** predicts the case marker on the subject and the object.

**Verb Phrase Classifier (Vp):** is used to predict the inflections on the verbs.

### 4.1 Subject and Object Classifier

Subject and Object phrases, when translated from English into a morphological rich language, often contain inflections of gender and number. Some languages also generate a case marker to denote the subject or the object. If such a case marker is not present, the target sentence often may not make sense. For our experiments from English to Hindi, we looked at predicting the correct case marker. To obtain the possible case markers that can come after a subject or the object in target language (in our case Hindi), we look at all the case markers following a subject and those that follow the object. If a language has linguistic resources such as parsers, this can be done easily. Since Hindi, and many other languages do not have a good parser, we make use of automatic word alignments obtained from bilingual data to project the subject information from English to Hindi, and determine the case markers following the subject and the object on the target side. Using this technique, we found 4 classes for the subject classifier and 3

<sup>1</sup>we use the libsvm library:  
<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

classes for the object classifier. For the prediction of the classes, we use all the noun phrase features (in Section 5.1), tense feature of the verb phrase (in Section 5.2) and tense conjugate features (in Section 5.3).

SubCM	ObjCM	Vp
NULL	NULL	<i>X raha tha</i> (was X+ing)
<i>ne</i>	<i>ko</i> (of)	<i>X+nA chahiye</i> (should X)
<i>ke</i> (of)	<i>mein</i> (in)	<i>X+nI chahiye</i> (should X)
<i>ki</i> (of)		<i>X+A gayA</i> (was X+ed) ...

Table 1: Classes defined for different classifiers.

## 4.2 Verb Phrase Classifier 1

Verb phrases contain morphological information about the gender, number, person, tense and aspect of the sentence. It is hence important to produce the right inflections and auxiliary verbs. Since it is impractical to have a class for each verb, we convert the verb phrases to an abstract form and also predict the target verb phrase in its abstract form. For instance, the verb phrase ‘was playing’ will be generalized to ‘was X+ing’ form and the corresponding predicted class would be ‘*X raha tha*’.

A simple approach to find the possible output forms of the classifier is to mine the target language data for all the verb phrases, rank them by frequency and filter them based on a threshold to yield the different forms that the verbs can take in the language. The aggregated verb phrases can be normalized by replacing the root verb in these phrases by an ‘X’ tag to obtain the possible abstract forms for the target verb phrases. For Hindi, verb phrases were identified by using a simple part-of-speech (POS) tagger to tag the monolingual data and to capture continuous sequences of ‘V’ tags. We found 120 Hindi verb classes in all. Some of these classes are listed in Table 1. We use all the features listed in Section 5.

## 4.3 Verb Phrase Classifier 2

Having too many classes for verb phrases causes the following problems: a) During our initial experiments we found that out of the 120 verb classes specified by us, only 60 were present in the bilingual training data. This reduces the chances of predicting a correct class since the classifier does not see all classes during training. b) The classifier sees only a few instances of each class. To simplify the verb phrase prediction, we split the prediction such that instead of predicting each verb form, we predict each ‘kind’ of inflection

that modifies the verb phrase. Since each verb phrase in our training data contains information about the gender, number and person, each class now has ample amount of training examples.

**Gender Classifier (VpG):** This classifier predicts the gender inflections on the target verb phrases using features from the source sentence.

**Number Classifier (VpN):** This classifier predicts the number inflections on the target verb phrases using features from the source sentence.

**Person Classifier (VpP):** This classifier predicts the Person information of the target verb phrases given the source sentence features.

The three classifiers have two, two and three classes, respectively. The predicted gender, number and person is then used to select the target verb form:

**Base Verb form Function:** Given the input English verb phrase, this function outputs all possible translations (that is, with all possible inflections and auxiliary verbs) of the given verb form. For example, for the verb phrase ‘is playing’ in the example in Section 1, this function will produce 12 target verb forms, one each for possible combinations of elements from the sets (masculine and feminine), (singular and plural) and (first, second and third person). The function for producing the list of verb forms given the English verb form is implemented using machine alignments and monolingual data as done in Gandhe et al. (2011). It uses parallel data to extract all the source-target verb phrase-pairs from the word-aligned data. These source-target verb phrase-pairs are converted into an abstract form by replacing the root verb with an ‘X’ (as done in Section 4.2). Aggregating this over a large amount of parallel data and filtering out the low frequency phrase-pairs gives us translations of a source verb form into its corresponding target forms. The gender, number and person for each of the target verb forms can be found out by looking at the inflections, suffixes and auxiliary verbs.

## 4.4 Training

We use an English parser to parse the source sentence and obtain the different features. Using the alignments of the subject, object and verb phrase,

we project them onto the target language and extract the expected output case-marker/inflections for each of the three cases (SubCM, ObjCM, Vp) and assign it the corresponding class. Our approach is not limited to hand-alignments. Alignments obtained from automatic aligners can also be used. Since hand-alignments were available beforehand, we made use of these alignments in this work. We will explore the usability of automatic aligners as future work. We now briefly describe the features that we used for the above classifiers.

## 5 Features

Given the parse tree of an English sentence, we determine the subject noun phrases and the object noun phrases for each of the verb phrases present in the input sentence giving *(subject, object, verb)* triples. We also determine the morphological information about the subject, object and verb phrases in sentence (in Sections 5.1 and 5.2). Most of the features described are boolean, unless specified otherwise. Figure 1 shows an example of an English-Hindi word-aligned sentence-pair. The dependency parse of the English sentence is used to determine the source subject (sita), object(chess) and the verb phrase (is playing). Features are calculated over these phrases and the target words aligned to them in the word alignments are used to create the training examples for the three classifiers.

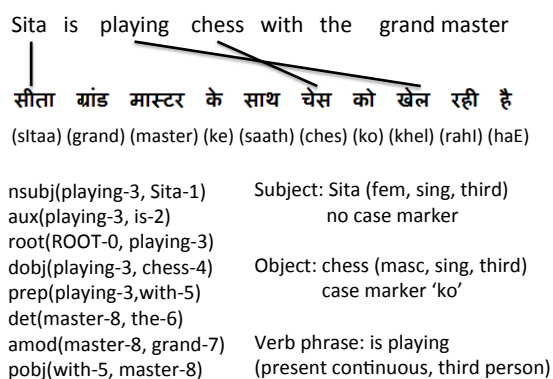


Figure 1: An English parse with features.

### 5.1 Noun Phrase Features

The inflection on the verb phrase is influenced by 3 attributes of a noun phrase:

**Gender:** Unlike English, most Indian languages have a gender (*male/female*) for every

subject and object. To determine the gender of an English word, we take its most common Hindi translation and assign the gender of this translation to the English word. Gender of Hindi words can be determined by mining the Hindi monolingual data for *(noun phrase, verb phrase)* pairs using a simple POS tagger on Hindi data. POS taggers are now easily available for most Indian languages. However, no other rich sources such as, parsers or morphological analyzers are used on the target language. We then assign the gender of the verb phrase suffix (*'a'* for masculine and *'I'* for feminine) to the words in the noun phrase. Doing this over a large amount of data gives us the list of nouns with their gender. For example, the Hindi word *'kItAb'* is seen with verb phrases such as, *'padI'*, *'dI'*, etc. in the monolingual data. Since *'kItAb'* occurs most with verb phrases ending in suffix *'I'*, its gender is *'female'*. The English word *'book'* translates most often to *'kItAb'* and is hence assigned the gender *'female'* and the corresponding feature value of 1. For words like, *'house'*, which are determined to be *'male'*, the value is 0.

**Number:** Similar to the gender, the singularity or plurality of the noun phrase influences the inflection on the verb phrase. The plurality of the English noun can be determined by using a POS tagger and looking for a *'NNS'* tag or in case of pronouns, a finite list of pronouns. Hence, nouns in plural form and the pronouns, *'they'*, *'us'*, *'them'*, were given the feature value as 1. For all other singular words, the value is 0.

**Presence of case marker:** Perhaps the most important feature, the presence or absence of a case marker on the target subject and object phrase decides the transfer of inflections from the noun phrases to the verb phrase (examples of Section 2). This is not a source side feature, since case markers are present on the noun phrases in the target language. We cannot use the case marker information directly as we do not have the target side information. Hence they are used in two steps: a) Subject and Object classifiers (Section 4) are used to predict the noun phrase (*subject, object*) case markers and b) The predicted case markers are used as an input to the verb phrase classifier. This feature is not used as an input to the subject and object classifiers. If a

subject/object case marker is present, the features are valued 1, else 0.

## 5.2 Verb Phrase Features

The verb phrase features influence the tense, aspect and person of the target verb phrase as well as the case marker presence on the noun phrases. The verb phrase extracted from the dependency parse of the input sentence are morphologically segmented (Minnen et al., 2001) and the different aspects of the verb phrase are obtained from it.

**Tense Features:** The tense features tell the presence or absence of *Present, Past and Future* tense. For instance, for the verb phrase ‘was explained’, the present and future features take the value 0 and the past feature takes the value 1.

**Aspect Features:** The aspect features are important in deciding the final form and the auxiliaries in the target sentence. We label the features as *simple, progressive and perfect*. In this case, a verb phrase with a ‘ing’ suffix is said to be progressive, whereas a verb phrase with ‘have’ and its inflections is said to be perfect. For example, the phrase ‘has been explaining’ will have both progressive and perfect features with value 1.

**Mood Features:** The mood features capture the *obligation, conditional and probability* mood in the input English sentence by looking at the modal verbs which are required to produce the corresponding auxiliary verbs in Hindi.

**Number:** English verb forms with plurality inflection translates into plurality of the Hindi verbs.

**Person:** English auxiliary verb ‘am’ denotes the presence of first person. By looking at the subject of the verb in the dependency parser, (*first, second or third*) the person information can be assigned to the verb phrase.

## 5.3 Conjugate Features

These features capture the more language-specific nuances that together decide the transfer of inflections from nouns to verbs. These features try to emulate the behavior of grammar rules.

**Case marker-Gender:** When a case marker is not present on the noun phrase, the inflection from

them is likely to be transferred to the verb phrase. For this case, we assign this feature the same value as the gender of the noun phrase. When a case marker is present, information is blocked and hence we assign a null value to this feature.

**Case marker-Number:** This feature captures blocking of the number information and takes a value 0 or 1 depending on the presence or absence of case marker.

**Tense-Gender:** When the tense of the sentence is past, it is likely that the gender information is blocked. Hence, when the tense is past, this feature is assigned a null value. Otherwise, the value is same as the value of the gender feature.

**Tense-Number:** Similar to the previous one, except that this captures the blocking of number information.

## 6 Decoding

We used a PBSMT system, similar to Tillman et al. (2006), to decode and this required slight modifications to incorporate our approach. The extracted phrase-pairs have phrase translation probabilities and lexical probabilities estimated (similar to Papineni et al. (2002)). The input sentence is passed through a parser to determine the subject, object and the verb phrases in the sentence. Various features mentioned in the previous section are computed during run time and the classifiers are used to predict the subject case marker, object case marker and the verb phrase inflection. The agreement constraints can be applied as:

**Hard Removal:** All phrase-pairs that do not agree with the predicted case marker or inflections are removed from the phrase table before the hypothesis search.

**Soft Removal:** The agreement model outputs the prediction probabilities for different target case markers or inflections. This probability score can be used as a feature in the phrase table and trained on a development data set.

**Addition:** If the predicted case marker or inflection is not present in the original phrase table, the correct phrase-pair can be added by

automatically generating the target phrase.

The input sentence is fed into the agreement model to produce the constraints for the subject, object and verb phrases. We use the hard constraint and addition techniques during decoding. Applying soft constraints will be done in future work. For subject and object phrases, we aggregate the phrase-pairs in the phrase table which contain the English source word. From these, all phrase-pairs that do not agree with the predicted case markers on the target side are filtered. In addition, if the predicted case marker is not present in the phrase table, we add the phrase-pair with the right case marker into the phrase table. This is done by looking for the most common target translations of the source word and appending the predicted case marker to them. For verb phrases, we aggregate the phrase-pairs containing the English verb phrase.

All phrase-pairs which do not have the predicted target verb phrase inflections are filtered. Since we do not know the complete translation of the source verb phrase at this step, we look only for the predicted target verb phrase's inflection and auxiliary verbs. If no correct verb phrase form is found in the phrase table, the target phrase is generated using the most common translation of the English verb and the phrase-pair is added. In order to score these new phrase-pairs, we can make use of the automatically generated bilingual dictionaries created during the automatic word-alignment phase. The phrase-pairs and entries in the dictionaries can be stemmed to their base forms (removing inflections) using Ramanathan et al. (2003). In cases where there are multiple instances of the same verb (caused due to stemming) present in the modified dictionary, the average of the probabilities is taken. The lexical probabilities for the phrase-pairs can then be estimated as given in Papineni et al. (2002) from the modified dictionaries. To obtain the phrase translation probabilities, the scores from the classifiers are converted to a score between 0 and 1 using a logistic function ( $1/(1 + e^{-score})$ , where, *score*: classifier's score) and then re-normalized such that the sum of probabilities of all the target phrases for a particular source phrase is one (and vice versa). In the case of 'Verb Phrase Classifier 2' (Section 4.3), the scores from each of the classifiers is first converted to a score between 0 and 1 using a logistic function, summed and then re-normalized.

## 7 Experiments

We first report the results of prediction of noun phrases and verb phrases and proceed on to report the results of using them in PBSMT.

### 7.1 Prediction Evaluation

To aggregate the classes required for subject, object and verb phrase classifiers, we used 1.4 million Hindi monolingual sentences crawled from the web. We pos-tagged this data using iit kgp Hindi pos tagger<sup>2</sup>. The monolingual data, along with 280,000 automatic alignments of sentence-pairs, was used to apply the technique suggested in Gandhe et al. (2011) to build the base verb form function described in Section 4.2. The svm classifiers were trained and tested using libsvm<sup>3</sup>. To extract the features from manually aligned sentences, we used the Stanford Parser<sup>4</sup> to obtain the English dependency parse trees. The source English side was morphologically segmented using morpha (Minnen et al., 2001) and the target Hindi side was segmented using an approach described in Ramanathan et al. (2003).

Table 2 gives the accuracies of the classifiers when trained with a particular set of features. The conjugate features make a significant improvement to all the three classifiers. Hindi object case markers are easier to predict than subject case markers since the objects usually do not occur with a case marker. Also, the subject case markers show a high dependency on the verb phrase features, which is explained by grammatical rules, according to which tense and structure of the verb phrase decide the case marker on the subject. It is important to remember here that the verb phrase classifier uses the output of the case-markers predicted by noun classifiers as a feature.

Features	SubCM	ObjCM	Vp
NounFeat	0.63	0.81	-
Noun+VerbFeat	0.72	0.84	0.58
Noun+Verb+ConjFeat	<b>0.75</b>	<b>0.87</b>	<b>0.61</b>

Table 2: Prediction accuracy for the classifiers.

The prediction accuracy is low for the **Vp** classifier even with conjugate features due to the large number of classes. Most classes do not have sufficient training examples and a few classes were

<sup>2</sup><http://nltr.org/snltr-software/>

<sup>3</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

<sup>4</sup><http://nlp.stanford.edu/software/lex-parser.shtml>



even absent in the training data. When we split this classification into separate tasks as explained in Section 4.3 and later combine the output of individual classifiers to obtain the predicted verb phrase, we obtain a much better accuracy. The results of this configuration are shown in Table 3. Since the verb phrase classifier uses case-markers as a feature, we also analyze the importance of these for verb phrase prediction and study 3 different settings: a) Removing the case marker (CM) feature, b) Using Gold case markers from the reference and c) Using the predicted case markers. Although the prediction accuracies are best for GoldCM, using the predicted case markers results in only a slight drop in accuracy.

	VpN	VpG	VpP	Overall
No CM	0.83	0.62	0.95	0.58
Gold CM	<b>0.87</b>	<b>0.86</b>	<b>0.95</b>	<b>0.74</b>
Pred CM	0.85	0.83	0.95	0.70

Table 3: Prediction accuracy for verb phrase inflections.

## 7.2 Machine Translation Evaluation

The system was trained on 285,000 automatically aligned sentences. The baseline system uses the standard decoding algorithm while our approach prunes the phrase table before decoding. We measure the translation quality using a single reference BLEU (Papineni et al., 2002). The test set contains 715 sentences from the News domain. Table 4 gives the comparison of the baseline with the two systems (Note: In both systems, the case marker features are obtained from the predictions of the subject and object classifiers):

**Pred1:** Verb phrase prediction as a single task (Table 2)

**Pred2:** Verb phrase prediction split into individual components (Table 3).

	BLEU	Adequacy	Fluency
Baseline	15.43	3.75	2.23
Pred1	15.45	3.87	2.41
Pred2	<b>15.58</b>	<b>3.93</b>	<b>2.79</b>

Table 4: BLEU score and Human Judgment.

The BLEU score increase is small on Pred1 but was significantly better with Pred2 with  $p < 0.0001$  with the Wilcoxon Signed-Rank test (Wilcoxon, 1945) performed by dividing the test file into 10 equal subfiles (as done in Gangadharaiah et al. (2010)). On analysis of the refer-

ence, we found the tense of the verb phrases in the Hindi reference to be different from that of English. Also, often the presence of auxiliary verbs ‘hona’ in the Hindi reference changed the structure of the verb phrase. The output produced by our system is more literal and in congruence with the grammar of the input sentence. Callison et al. (2006) list the disadvantages of using BLEU. The differences in translations between the proposed approaches and the baseline are most often a correction of inflection, and sometimes this resulted in better selection of neighboring words by the language model. BLEU failed to accommodate these improvements, hence we also performed human evaluation to judge the quality of the translations on adequacy and fluency using a scale of 1-5<sup>5</sup>.

We gave 100 randomly picked sentences from the test set to a single human judge. We see that our approach (Table 4) has a greater impact on fluency, suggesting that grammatical agreement is important for fluency. Adequacy improvement can be attributed to the correct translations of the case markers and the tense information.

## 8 Conclusion and future work

We modeled the task of case marker and inflection prediction as a classification task. The prediction accuracies show that the inflections on the verbs are highly influenced by the case markers on the subjects and objects. Similarly, the case markers on subjects are affected by the tense of the verb phrases. Since all the features are extracted from the source side, this approach can be easily applied for improving translation quality from English to any morphologically rich foreign language. More work can be done on creating features that encode the grammatical rules we might have missed.

Even though the gain in translation quality with the BLEU score was small, human evaluation showed that this approach helps in improving the fluency and adequacy of the sentence and hence makes it more readable. Future work can be on using more than one possible case marker-verb phrase constraints (i.e., as a soft constraint) for a given input and applying this approach for other language-pairs where the target language is morphologically rich.

<sup>5</sup>We used the scale defined in <http://projects.ldc.upenn.edu/TIDES/Translation/TransAssess04.pdf>

## References

- C. Callison-Burch, M. Osborne, and P. Koehn. 2006. Re-evaluating the role of bleu in machine translation research. In *EACL*, pages 249–256.
- K. Crammer and Y. Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. In *Journal of Machine Learning Research*, pages 265–292.
- A. T. Freeman, S. L. Condon, and C. M. Ackerman. 2006. Cross linguistic name matching in english and arabic: a “one to many mapping” extension of the levenshtein edit distance algorithm. 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 471–478.
- A. Gandhe, R. Gangadharaiah, K. Visweswariah, and A. Ramakrishnan. 2011. Handling verb phrase morphology for indian languages in machine translation. In *Proceedings of the International Joint Conference on Natural Language Processing*. Asian federation for NLP.
- R. Gangadharaiah, R. D. Brown, J. Carbonell. 2010. Monolingual distributional profiles for word substitution in machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 320–328.
- S. Green and J. DeNero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL, pages 146–155.
- N. Habash. 2008. Four techniques for online handling of out-of-vocabulary words in arabic-english statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short ’08, pages 57–60.
- P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, F. Marcello, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics (ACL)*, demonstration session.
- A. Lavie. 2008. Stat-xfer: a general search-based syntax-driven framework for machine translation. In *Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, CICLing, pages 362–375.
- E. Minkov, K. Toutanova, and H. Suzuki. 2007. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL, pages 128–135.
- G. Minnen, J. Carroll, and D. Pearce. 2001. Applied morphological processing of english. In *Natural Language Engineering*.
- K. Papineni, S. Roukos, T. Ward and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pages 311–318.
- M. Popovic and H. Ney. 2004. Towards the use of word stems and suffixes for statistical machine translation. In *Proceedings of The International Conference on Language Resources and Evaluation*.
- A. Ramanathan and D. Rao. 2003. A Lightweight Stemmer for Hindi. *Workshop on Computational Linguistics for South-Asian Languages*, EACL.
- A. Ramanathan, H. Choudhary, A. Ghosh and P. Bhattacharyya. 2009. Case markers and Morphology: Addressing the crux of the fluency problem in English-Hindi SMT. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 800–808.
- C. Tillman. 2006. Efficient Dynamic Programming Search Algorithms for Phrase-based SMT. In *Proceedings of the Workshop CHPSLP at HLT’06*.
- K. Toutanova, H. Suzuki, and A. Ruopp. 2008. Applying morphology generation models to machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 514–522.
- F. Wilcoxon. 1945. *Individual comparisons by ranking methods*. *Biometrics*, 1, 80-83, <http://faculty.vassar.edu/lowry/wilcoxon.html>.
- P. Williams and P. Koehn. 2011. Agreement constraints for statistical machine translation into german. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT, pages 217–226.
- M. Yang and K. Kirchhoff. 2006. Phrase-based back-off models for machine translation of highly inflected languages. In *Proceedings of the 21st International Conference on Computational Linguistics*, pages 1017–1020.

# Scalable Variational Inference for Extracting Hierarchical Phrase-based Translation Rules\*

**Baskaran Sankaran**  
Simon Fraser University  
Burnaby BC, Canada  
baskaran@cs.sfu.ca

**Gholamreza Haffari**  
Monash University  
Clayton VIC, Australia  
reza@monash.edu

**Anoop Sarkar**  
Simon Fraser University  
Burnaby BC, Canada  
anoop@cs.sfu.ca

## Abstract

We present a Variational-Bayes model for learning rules for the Hierarchical phrase-based model directly from the phrasal alignments. Our model is an alternative to heuristic rule extraction in hierarchical phrase-based translation (Chiang, 2007), which uniformly distributes the probability mass to the extracted rules locally. In contrast, in our approach the probability assigned to a rule is globally determined by its contribution towards all phrase pairs and results in a sparser rule set. We also propose a distributed framework for efficiently running inference for realistic MT corpora. Our experiments translating Korean, Arabic and Chinese into English demonstrate that they are able to exceed or retain the performance of baseline hierarchical phrase-based models.

## 1 Introduction

Hierarchical phrase-based translation (Hiero) as described in (Chiang, 2005; Chiang, 2007) uses a synchronous context-free grammar (SCFG) derived from heuristically extracted phrase pairs obtained by symmetrizing bidirectional many-to-many word alignments (Och and Ney, 2004). The phrase-pairs are constrained by the source-target alignments such that all the alignment links from the source (target) words are connected to the target (source) words *within* the phrase. Given a word-aligned sentence pair  $\langle f_1^j, e_1^I, A \rangle$ , where  $A$  indicate the alignments, the source-target sequence pair  $\langle f_i^j, e_{i'}^{j'} \rangle$  can be a phrase-pair *iff* the following alignment constraint is satisfied.

$$(k, k') \in A : k \in [i, j] \Leftrightarrow k' \in [i', j']$$

Given the phrase-pairs, SCFG rules are extracted by replacing aligned sequences of words in source

\*This research was partially supported by an NSERC, Canada (RGPIN: 264905) grant and a Google Faculty Award to the third author.

and target sides by co-indexed non-terminals and rewriting the replaced source-target word sequences as separate rules. Consider a rule  $X \rightarrow \langle \beta, \gamma \rangle$ , where  $\beta$  and  $\gamma$  are sequences of terminals and non-terminals. Now, given another rule  $X \rightarrow \langle f_i^j, e_{i'}^{j'} \rangle$ , such that  $f_i^j$  and  $e_{i'}^{j'}$  are contained fully within  $\beta$  and  $\gamma$  as sub-phrases, the larger rule could be rewritten to create a new rule.

$$X \rightarrow \langle \beta_p X_k \beta_s, \gamma_p X_k \gamma_s \rangle \quad (1)$$

Here  $\beta_p$  ( $\beta_s$ ) refers to any prefix (suffix) of  $\beta$  that precedes (follows)  $f_i^j$ . Note that the non-terminals are co-indexed with a unique index so that they are rewritten simultaneously.

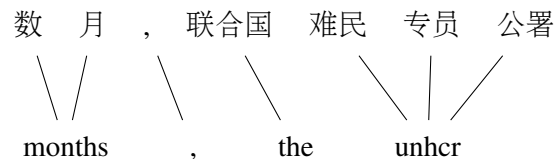


Figure 1: Chinese-English *phrase-pair* with alignments

As a concrete example, consider the word aligned Chinese-English phrase pair shown in Figure 1. Notice that the phrase 联合国 (united nations) is incorrectly aligned to English determiner *the*, even though in ideal case the entire Chinese phrase 联合国难民专员公署 should be aligned on the English side to *the unhr*. The heuristic approach extracts 32 rules, some of which are shown in Figure 2.

The distribution of the rules is unknown, as the different derivations of the sentences are not explicitly observed. Thus, Chiang (2007) follows an approach similar to that of Bod (1998) and hypothesizes a distribution over the rules. Under this each phrase-pair is assumed to have a unit count, which is uniformly distributed to all the rules extracted from this phrase-pair. The locally assigned rule counts are then aggregated across the entire set of phrase-pairs. The probability for each phrase pair

Translation rule	$P_{Heu}(e f)$	$P_{VB}(e f)$
* $X \rightarrow \langle \text{在中南海} \mid \text{at chong nan hai} \rangle$	0.333	0.003
* $X \rightarrow \langle \text{在中南海} \mid \text{at zhong nan hai} \rangle$	0.333	0.008
$X \rightarrow \langle \text{在中南海} \mid \text{at zhongnanhai} \rangle$	0.333	<b>0.988</b>

Figure 3: Rules extracted for translating the Chinese phrase 在中南海. The probabilities are shown for grammar extracted from heuristic as well our proposed method. The least preferred translations are shown with \*. Our Variational-Bayes method extracts a grammar having a peaked distribution as shown.

* $X \rightarrow \langle \text{联合国} \mid \text{the} \rangle$
* $X \rightarrow \langle \text{数月, 联合国} \mid \text{months, the} \rangle$
$X \rightarrow \langle \text{数月, } X_1 \mid \text{months, } X_1 \rangle$
$X \rightarrow \langle \text{数月} \mid \text{months} \rangle$
$X \rightarrow \langle \text{联合国难民专员公署} \mid \text{the unhcr} \rangle$

Figure 2: Rules extracted for the example phrase-pair in Figure 1. The rules encoding incorrect translations are marked with \*.

is then estimated using relative frequency estimation.

### 1.1 Motivation

A major problem with this heuristic rule extraction method is the lack of *global re-weighting* of the pseudo-counts beyond their local assignments. By assigning uniform weight to the rules, Chiang (2007) assumes all the rules extracted from a given phrase-pair to be equally good. However, some rules might be better than others in terms of generalization, for capturing a syntactic phrase-pair, or being a semantically coherent unit of translation.

Due to this uniform treatment of good and poor translations, probability mass is wasted on poor translation candidates. For example the phrase-pair in Fig 1 would generate several poor translation rules (shown with \* in Fig. 2). This is due to the incorrect alignment link between 联合国 and *the* (note that the word *the* is typically aligned with a large number of words due to its frequency). The heuristic extraction method simply assigns uniform count to all translations and as a result the first translation in Fig. 2 becomes the fourth best translation for this source phrase.

In Chiang (2007) the rule extraction algorithm produces a fairly flat distribution over rules. For example the different translation options of the Chinese phrase 在中南海 (*at zhongnanhai*) all

have the same  $p(e|f)$  probability as shown in Figure 3. In contrast, our method produces a peaked distribution and shifts the probability mass towards *at zhongnanhai*, which is the preferred translation.

In this paper, we propose a method which distributes the probability mass among the rules (generated from a phrase-pair) based on their contribution in explaining the collection of all phrase-pairs in a global manner. This difference in estimation methods can lead to a peaked distribution of rule probabilities. Secondly we also present a distributed framework that enables rule extraction on large datasets that are typical in SMT. Our Variational-Bayes approach for rule extraction improves/ retains the translation quality for the three different language pairs. Finally, we also present a detailed analysis comparing the extracted SCFG with the heuristically extracted SCFG.

## 2 Model

Our model uses the notion of a *derivation*: the set of rules that fully derive an aligned phrase pair, and learns the estimates for the rules contained in the derivations through Variational inference. Setting the notations, we denote the set of derivations for a given phrase pair  $x$  as  $\phi_x$  and the set of all rules as  $\mathcal{G}$ . Given the set of initial phrase pairs  $\mathcal{X}$  and a prior over the grammars  $\mathcal{G}$ , we formulate Hiero grammar extraction as the task of inferring a posterior distribution over Hiero grammars. Using Bayes' rule, we can express the posterior over the grammar  $\mathcal{G}$  given the set of bilingual phrases  $\mathcal{X}$  as:  $P(\mathcal{G}|\mathcal{X}) \propto P(\mathcal{G})P(\mathcal{X}|\mathcal{G})$ .

As mentioned earlier, our model replaces the heuristic rule extraction step in Hiero pipeline. Consequently our model assumes the existence of *initial* phrase-pairs obtained from bidirectional symmetrization of word alignments. We use the following two-step generative story to create an aligned phrase pair from the Hiero rules.

$\phi^z \sim \text{Dirichlet}(\alpha_z)$	[draw derivation type parameters]
$\theta \sim \text{Dirichlet}(\alpha_h p_0)$	[draw rule parameters]
$z_d \sim \text{Multinomial}(\phi^z)$	[decide the derivation type]
$r \mathbf{r} \in d_x \sim \text{Multinomial}(\theta)$	[generate rules deriving phrase-pair $x$ ]

Figure 4: Definition of the proposed model

1. First decide the derivation type  $z_d$  for generating the aligned phrase pair  $x$ . It can either be a terminal derivation or hierarchical derivation with one/two gaps,<sup>1</sup> i.e.  $z_d = \{\text{TERM}, \text{HIER-A1}, \text{HIER-A2}\}$ .
2. Then identify the constituent rules  $\mathbf{r}$  in the derivation to generate the phrase pair.

Under this model the probability of a particular derivation  $d \in \phi_x$  for a given phrase pair  $x$  can be expressed as:

$$p(d) \propto p(z_d) \prod_{r \in d} p(r|\mathcal{G}, \theta) \quad (2)$$

where  $r$  is a rule in grammar  $\mathcal{G}$  and  $\theta$  is the grammar parameter.

Figure 4 depicts the generative story of our generative model. The derivation-type  $z_d$  is sampled from a multinomial distribution parameterized by  $\phi^z$ , where  $\phi^z$  is distributed itself by a Dirichlet distribution with hyper-parameter  $\alpha_z$ . The grammar rules are generated from a multinomial distribution parameterized by  $\theta$ , where  $\theta$  itself is distributed according to a Dirichlet distribution parameterized by a concentration parameter  $\alpha_h$  and a base distribution  $p_0$ . For the base distribution, we use a simple but yet informative prior based on geometric mean of the bidirectional alignment scores. This allows us to only explore the rules that would be consistent with the underlying word alignments.<sup>2</sup> Thus our setting closely resembles that of the Hiero heuristic rule extraction.

Our goal is thus to infer the joint posterior  $p(\theta, \Phi|\alpha_h, p_0, \alpha_z, \mathcal{X})$ , where  $\theta$  are the model parameters and  $\Phi$  the latent derivations over all the phrase pairs.

<sup>1</sup>This refers to the maximum arity of a rule involved in the derivation.

<sup>2</sup>While a non-parametric prior would be better from a Bayesian perspective, we leave it for future consideration.

### 3 Training

For inference we resort to a variational approximation and factorize the posterior distributions over grammar parameters  $\theta$  and latent derivations  $\Phi$  as:

$$p(\theta, \Phi|\alpha_h, p_0, \alpha_z, \mathcal{X}) \approx q(\theta|\mathbf{u})q(\Phi|\pi)$$

where  $\mathbf{u}$  and  $\pi$  are the parameters of the variational distributions.

The inference is then performed in an EM-style algorithm- iteratively updating the parameters  $\mathbf{u}$  and  $\pi$ . We initialize  $\mathbf{u}^0 := \alpha_h p_0$ , which is then updated with expected rule counts in subsequent iterations. The expected count for a rule  $r$  at time-step  $t$  can be written as:

$$\mathbb{E}[r^t] = \sum_{d \in \phi_x} p(d|\pi^{t-1}, x) f_d(r) \quad (3)$$

where  $p(d|\pi^{t-1}, x)$  is the probability of the derivation  $d$  for the phrase pair  $x$  and  $f_d(r)$  is the frequency of the rule  $r$  in derivation  $d$ . The  $p(d|\cdot)$  term in Equation 3 can then be written in terms of  $\pi$  as:

$$p(d|\pi^{t-1}, x) \propto p(z_d) \prod_{r \in d} \pi_r^{t-1} \quad (4)$$

The  $p(d|\cdot)$  are normalized across all the derivations of a given phrase pair to yield probabilities. For each *derivation type*  $z_d$ , its expected count (at time  $t$ ) is the sum of the probabilities of all the derivations of its type.

$$\mathbb{E}[z_d^t] = \sum_x \sum_{\{z_d=z_{d'}|d' \in \phi_x\}} p(d'|\pi^{t-1}, x) \quad (5)$$

We initialize the Dirichlet hyperparameters  $\alpha_{z_d}$  using a Gamma prior ranging between  $10^{-1}$  and  $10^3$ :  $\alpha_{z_d} \sim \text{Gamma}(10^{-1}, 10^3)$ .<sup>3</sup>

<sup>3</sup>In initial experiments we used an initial prior of  $\alpha_z = [10^0, 10, 10^4]$  to compensate for the smaller probabilities for arity-2 derivation resulting from two multiplications. However, our later experiments showed it to be unnecessary and so we used an initial prior that does not prefer any particular outcome.

---

**Algorithm 1** Variational-Bayes for Hiero Rules

---

**Input:** Set of aligned phrase-pairs  $\mathcal{X}$   
Get prior distribution  $\mathbf{u} = \{u_r = \alpha_h p_0(r) | r \in \mathcal{G}\}$   
Set  $\mathbf{u}^0 = \mathbf{u}$   
**for** time-step  $t = 1, 2, \dots$  **do**  
  **for**  $z_d \in Z$  **do**  
     $p(z_d) \leftarrow \exp\left(\psi(\alpha_{z_d}^{t-1}) - \psi(\sum_{z_d} \alpha_{z_d}^{t-1})\right)$   
  **end for**  
  **for**  $r \in \mathcal{G}$  **do**  
     $\pi_r^{t-1} \leftarrow \exp\left(\psi(u_r^{t-1}) - \psi(\sum_r u_r^{t-1})\right)$   
  **end for**  
  **for**  $x \in \mathcal{X}$  **do**  
    **for**  $d \in \phi_x$  **do**  
      Compute  $p(d|\pi^{t-1}, x)$  as in (4)  
       $\mathbb{E}[z_d^t] \leftarrow \mathbb{E}[z_d^t] + p(d|\pi^{t-1}, x)$   
      **for**  $r \in d$  **do**  
         $\mathbb{E}[r^t] \leftarrow \mathbb{E}[r^t] + p(d|\pi^{t-1}, x) f_d(r)$   
      **end for**  
    **end for**  
  **end for**  
  **for**  $z_d \in Z$  **do**  
    Estimate  $\alpha_{z_d}^t \leftarrow \alpha_{z_d}^0 + \mathbb{E}[z_d^t]$   
  **end for**  
  **for**  $r \in \mathcal{G}$  **do**  
    Estimate posterior  $\mathbf{u}^t$ :  $u_r^t \leftarrow u_r^0 + \mathbb{E}[r^t]$   
  **end for**  
**Output:** Posterior distribution  $\mathbf{u}^t$

---

We run inference for a fixed number of iterations<sup>4</sup> and use the grammar along with their posterior counts from the last iteration for the translation table. Following (Sankaran et al., 2011), we use the shift-reduce style algorithm to efficiently encode the word aligned phrase-pair as a normalized decomposition tree (Zhang et al., 2008). The possible derivations (that are consistent with the word alignments) could then be enumerated by simply traversing every node in the decomposition tree and replacing its span by a non-terminal  $X$ .

### 3.1 Distributing Inference

While the above training procedure works well for smaller datasets, it does not scale well for the realistic MT datasets (which have millions of sentence pairs) due to greater memory and time requirements. To address this shortcoming, we distribute the training using a Map-Reduce style framework, where each node works on the local dataset in computing the required statistics and then communicates the statistics to a central aggregator reduce node.

Distributed inference for Expectation Maximization algorithm was studied in (Wolfe et al., 2008). They used three different topologies in

<sup>4</sup>In our experiments, we set the number of iterations to 10.

terms of computation time, bandwidth requirement and so on. While Map-Reduce is substantially slower than the All-pairs and Junction-tree topologies, it takes much lesser bandwidth than the other two apart from being much easier to implement. Furthermore our choice of the Variational inference naturally lends itself to distributed training.

We simply shard the set of aligned phrase pairs and parallelize the training steps for the shards across different nodes. At the end of local computation of the statistics (expected rule counts for example), we need to aggregate the statistics to get a global view, which will then be used in the next iteration/training step. We parallelize this aggregation across several nodes in one or two reduce steps as required. At the end of aggregation we communicate the updated statistics to each node on a need basis.<sup>5</sup>

## 4 Experiments

We experiment with three datasets of varying sizes. We use the University of Rochester Korean-English dataset consisting of almost 60K sentence pairs for the small data setting. For moderate and large datasets we use Arabic-English (ISI parallel corpus) and Chinese-English (Hong Kong parallel text and GALE phase-1) corpora. We use the MTC dataset having 4 references for tuning and testing for our Chinese-English experiments. The statistics of the corpora used in our experiments are summarized in Table 1.

Lang.	Training Corpus	Train/ Tune/ Test
<i>Ko-En</i>	URochester data	59218/ 1118/ 1118
<i>Ar-En</i>	ISI Ar-En corpus	1.1 M/ 1982/ 987
<i>Cn-En</i>	HK + GALE ph-1	2.3 M/ 1928/ 919

Table 1: Corpus Statistics in # of sentences

We follow the standard MT practice and use GIZA++ (Och and Ney, 2003) for word aligning the parallel corpus. We then use the heuristic step that symmetrizes the bidirectional alignments (Och et al., 1999) to extract the initial phrase-pairs up to a certain length, consistent with the word alignments. Finally we employ our proposed Variational-Bayes training to learn rules for

<sup>5</sup>We simulate the Map-Reduce style of computation using a regular high-performance cluster using a mounted filesystem rather than a Hadoop cluster with a distributed filesystem.

Model	Ko-En	Ar-En	Cn-En
<i>Baseline</i>	7.18	<b>37.82</b>	<b>28.58</b>
<i>Variational-Bayes</i>	<b>7.68</b>	37.76	28.40

Table 2: BLEU scores for baseline heuristic extraction and the proposed Variational-Bayes model. Best scores are in **boldface** and statistically significant differences are *italicized*.

Hiero. As a baseline Hiero model, we use the heuristic rule extraction (Chiang, 2007) approach to extract the rules. In both cases the parameters are estimated by the relative frequency estimation.

For decoding we use our in-house hierarchical phrase-based system- Kriya<sup>6</sup> (Sankaran et al., 2012b). We use the following 8 standard features for the log-linear model: translation probabilities ( $p(e|f)$  and  $p(f|e)$ ), lexical probabilities ( $p_l(e|f)$  and  $p_l(f|e)$ ), phrase and word penalties, language model and glue rule penalty.

#### 4.1 Results

The main BLEU score results are summarized in Table 2 and the key aspects are summarized below.

- **Higher BLEU scores:** Our Bayesian model performs better than the baseline heuristic rule extractor for Korean-English. Furthermore, the improvement of 0.5 BLEU is statistically significant at  $p$ -value of 0.01.
- **Large corpora:** Our distributed inference model easily scales to the large corpora and the inference completes in less than a day for Chinese-English. It also retains BLEU scores in the same level as the baseline models for both Arabic-English and Chinese-English.

#### 4.2 Compact Models

Some earlier research on Hiero have explored model size reduction as a means of reducing the time and space complexity of the Hiero decoder as well as for mitigating issues such as *overgeneration* (Setiawan et al., 2009). These approaches use a variety of compression strategies, *viz.* threshold pruning (Zollmann et al., 2008), pattern-based filtering (He et al., 2009; Iglesias et al., 2009) and significance pruning (Yang and Zheng, 2009).

While compact models is not the central idea of our work, we nevertheless explore the effect of

<sup>6</sup><https://github.com/sfu-natlang/Kriya>

a simple threshold pruning strategy on the grammar learned from our proposed model. Table 3 shows the results for the pruned grammars, where we prune the rules having expected count below a mincount threshold. We present the results for specific mincount settings based on our experiments on the held-out tuning set for each language pair.

Model	Ko-En	Ar-En	Cn-En
Model size: <i>VB</i>	2.67	331.6	471.7
BLEU: <i>VB</i>	<b>7.68</b>	<b>37.76</b>	<b>28.40</b>
<i>Pruning mincount</i>	0.25	1.0	1.0
Model size: <i>pruned</i>	1.65	58.9	87.3
Reduction:	38.2%	82.2%	81.5%
BLEU: <i>pruned</i>	<b>7.64</b>	37.58	<b>28.45</b>

Table 3: Model sizes (in millions) and BLEU scores of the full VB and pruned VB grammars. Mincount implies the expected rule count threshold used for pruning the full VB grammar. Best/indistinguishable BLEU scores are shown in **boldface**.

- **Retains score with smaller grammar:** The pruned grammars retain the performance of the full grammar, even while using just 18% of the complete model.
- **Higher reduction for large dataset:** Variational inference reduces the model size over 80% for the large corpora. While this is similar to the findings of Johnson et al. (2007) and that of the pruning strategies mentioned above; the question of whether an intelligent model selection strategy can yield higher BLEU scores is still open.
- **Faster decoding:** The compact grammars naturally result in faster decoding and we observed up to 20-30% speedup in the translation including the time spent for loading the model.

Sankaran et al. (2012a) proposed a model for extracting *compact* Hiero grammar with restricted arity (at most 1 non-terminal). In contrast our model is close the classical Hiero model Chiang (2007) having an arity of two. Though our results are not directly comparable to theirs, we nevertheless find our model to yield a better model size reduction than theirs. While they claim up to 57%

reduction, we achieve over 80% for the two large data conditions and about 38% reduction for the Korean-English small data setting.

### 4.3 Analysis

We now compare the probability distributions of the two grammars at the level of individual rules to understand the differences between them. We considered a set of source phrases that are common in both grammars and analyzed their probability distributions over the translation options.

Specifically we use the Q-Q plot to study the behaviour of two probability distributions as explained below considering the Chinese phrase 联合国 (*united nations*) as a representative example. The Q-Q plot in Figure 5 plots the  $p(e|f)$  probabilities (sorted for the baseline grammar) for different translations of the source phrase. The translations from the baseline grammar are then paired off with the points in the sorted VB curve and the corresponding probabilities are plotted in the same order as the baseline translations. The following conclusions can be drawn from this plot:

- **Penalize poor translations:** Among the low-probability translations, majority of the translations in the VB-grammar have probability less than the corresponding baseline translations. This has the desired property of potentially shifting the probability mass away from poor translations.
- **Reward good translations:** VB-grammar rewards some translations that were deemed to be poor by the heuristic method, by assigning a slightly higher probability than the heuristic grammar. A manual inspection showed that the rules with higher probabilities were objectively better translation rules. For example Table 4 contrasts the probabilities assigned by the two methods for the first four translation options in Fig. 5.
- **Uniform probability is not informative:** The heuristic extraction method tends to assign an uniform probability for groups of translations and this is evident in the flat segments of the baseline curve and is especially dominant in the low probability region. In contrast, the VB-grammar is more peaked (in Fig. 5 the probabilities are sorted for the VB grammar).

Translations	Heu $p(e f)$	VB $p(e f)$
alert the united nations	4.28e-04	3.46e-04
during	4.28e-04	3.20e-04
<i>for un</i>	4.28e-04	<i>5.02e-04</i>
human	4.28e-04	3.20e-04

Table 4: Probabilities assigned by the two methods for the first four translations in Fig. 5. The better translation among four and the higher probability assigned by our model are *italicized*.

We also observe similar trend for several source phrases in both Arabic-English and Chinese-English corpora.

At the macro level, we compare the sizes of the different types of rules in the heuristic and the Variational-Bayes grammar. The baseline grammar extract slightly more rules with arity-1 than the grammar extracted by our model (see Figure 6). Our model extracts rules used in a *derivation* of a phrase-pair, only if *all* its constituent rules are consistent with the Hiero rule constraints (such as restriction on the total number of terminals and non-terminals in the rule). However the heuristic method extracts all the consistent rules and does *not* consider the derivations. While this is a more stricter constraint, the VB model extracts slightly more (about 170K) arity-2 rules as we allow the unaligned words to be attached to different levels of hierarchical rules during the construction of the decomposition tree. This extracts translation rules that are beyond the purview of the heuristic method, since the Viterbi alignments cannot capture them.

We earlier examined the effect of pruning the VB grammar in Section 4.2 and noted that the grammar could be substantially reduced for different language pairs without sacrificing translation quality. In this context, we compare the effects of pruning the heuristic and VB grammars in Figure 6 for Chinese-English. For the same mincount threshold of 1 as the best performing VB setting, the BLEU score of the heuristic grammar drops by over 1 point. However this setting prunes over 99% of the arity-1 and arity-2 rules even while it retains all the terminal rules. This is primarily because of the way the heuristic method estimates rule counts by uniformly distributing the weight among all the rules. The terminal rules are sufficient for coverage but does not capture long distance movements; and the lack of arity-1 and arity-2 rules further restrict the reordering ability of the model. We have to substantially lower the



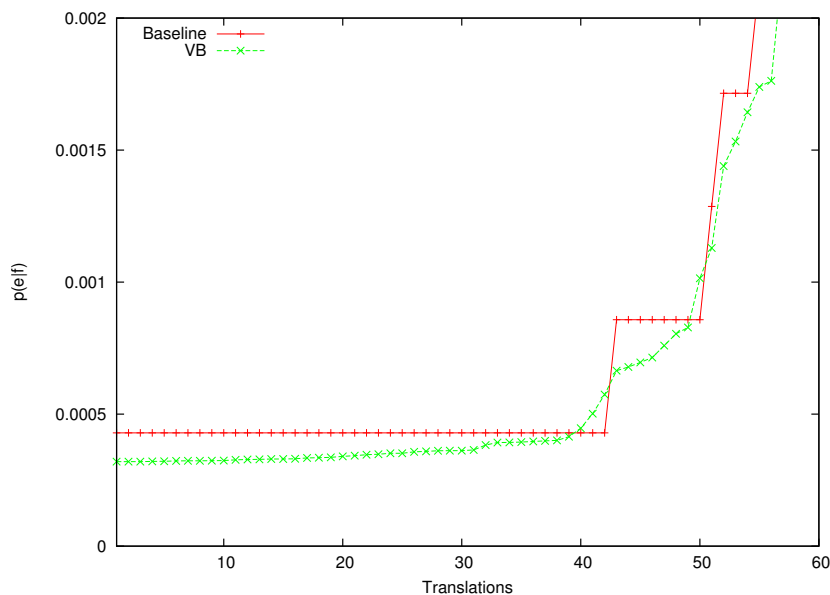


Figure 5: Q-Q plot comparing the  $p(e|f)$  distributions of the baseline and VB-grammars for the Chinese phrase 联合国 (*united nations*). The points on the two curves represent distinct target translations (the numbers on the  $x$ -axis indicate the indices) and the points are sorted according to VB translation probabilities against the paired-off translation probabilities from the baseline grammar. The  $y$ -axis is clipped to highlight the variations in the low-probability range.

mincount threshold to 0.05, in order to get performance comparable to the pruned VB grammar setting. Interestingly, this uses about 7M more rules (13M more arity-1 rules, but 6M fewer arity-2 rules) than VB-Pr (1.0), but its BLEU score is marginally lower than the latter. This could be ascribed to the missing arity-2 rules, which could be crucial for certain long-distance reordering.

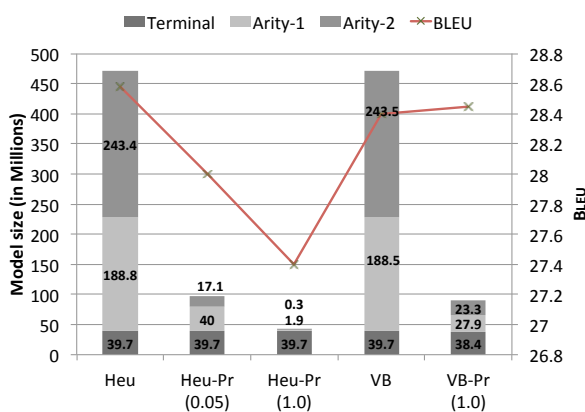


Figure 6: Cn-En: Model sizes and BLEU for different grammars. The pruned models are identified by the suffix 'Pr', whose mincount is shown in the brackets. The  $y$ -axis on the left marks the model sizes and that on the right denotes BLEU. The numbers in the stacked bars denote the # of rules (in millions) for the corresponding rule type.

As the final part of the analysis, we also present the 100 high probability lexical phrases extracted by both rule extraction methods for the Arabic-English corpus in Figure 7. As seen, the heuristic grammar assigns high probability to the rules translating proper nouns and short phrases, whereas the VB method assigns high probability to more generic translations.

## 5 Related Research

Most of the research on learning Hiero SCFG rules has been focussed on inducing phrasal alignments between source and target using Bayesian models (Blunsom et al., 2008; Blunsom et al., 2009; Levenberg et al., 2012; Cohn and Haffari, 2013). Broadly speaking, these generative approaches learn a posterior over parallel tree structures on the sentence pairs. While these methods extract hierarchical rules, they do not conform to Hiero-style rules. Consequently the hierarchical rules are used *only* for learning an alignment model and cannot be used directly in the Hiero decoder. Instead, these approaches employ the standard Hiero heuristics to extract rules to be used by the decoder from the alignments predicted by their model. In this sense, these are similar to Bayesian models for learning alignments using stochastic Inversion transduction grammars (ITG) (Wu, 1997) or linear

### Phrases from Heuristic Grammar

مليون سهم/million shares هوجو/hugo مؤشر الاسهم " ب " :/: b share index فيما يلي العناوين الرئيسية في/ in major news items وانديونيسيا ولاوس/ indonesia , laos , yuan ايجور/igor مؤشر الاسهم " ب " /b share index توني/tony جاك استرو/jack straw ، وفقا لما ذكره بنك/ national according to the bank of masood khan/مسعود خان/ manmohan singh/مانموهان سينغ/ khan jamali/خان جمالي/ jose hui/جوسيه هوي/ china ( hong kong )/الصين ( هونغ كونغ )/ security celta/سلتا/ nanjing/نانجينغ/ daniel kong/دانيل كونج/ kong ( hong kong )/الصين ( هونغ كونغ )/ kong kong/هونغ كونغ/ china ( hong kong )/الصين ( هونغ كونغ )/ athletico/أثليتيك/ china ( hong kong )/الصين ( هونغ كونغ )/ pradesh/براديش/ yoriko kawaguchi/يوريكو كاواغوتشي/ junichiro/جونيتشيرو/ jose/جوسيه/ yashwant/ياشوانت/ glafcos/جلافكوس/ chandrika/شانديريكا/ president eduardo duhalde/ادواردو دوهادلي/ herve de/هرفيه دو/ nato/ناتو/ nato/ناتو/ toledo/توليدو/ everton/يفرتون/ ramirez/راميريز/ social stability/الاستقرار الاجتماعي/ shares . /نهاية الخبر/ . / daniel/دانيل/ carlo/كارلو/ sepe/سيبي/ march 31/مارس 31/ kong/كونج/ kufuor/كوفور/ sese/سيسي/ turnover :/: اسعار بورصة طوكيو/ tokyo stock price/اللاتنطي / سارس/ syndrome/ sars الاصفر/ yellow نظيره الفلسطيني/ his palestinian counterpart/ 50/50 : جوزيه/ jose/الجارجية العراقي/ iraqi foreign/ وليتوانيا/ lithuania عبدالله جول/ abdullah كبار المتعاملين في الذهب في هونغ/ gold dealers in hong/ international humanitarian/ ين/ yen احد كبار المتعاملين في الذهب في هونغ/ one of the major gold dealers in hong/ fidel ساو تومي/ saو tome/ راسينغ/ racing اتليتيكو/ atletico بالصفة الغربية/ west bank ين (//) yen تشاو شينغ/ zhaoxing سيلفا/ silva فيردر/ werder فرناندو/ fernando امريكي (//) dollars الصيني الزائر/ visiting chinese/ 1 0 bank الخارجية الامريكية نيكلوس/ nicholas في عام 1996 . / in بغداد --/ baghdad الرئيس موسيني/ president museveni/ الرئيس فلاديمير/ vladimir the major/ هونغ/ hong زوران/ zoran يانغ لي وي/ yang liwei ين الي/ yen تو/ tokyo

### Phrases from Variational-Bayes Grammar

21 يونيو/21 june الشيخ علي/ sheikh ali 100 طن/100 tons البيت الابيض سكوت مكليلان/ scott mcclellan من 77/77 from دومنيك دو/ dominique de جيانغسو/ jiangsu المعلومات الالكترونية/ electronic information الأمريكي دونالد/ donald افريقيا 1/1 africa سانشا فرنسا/ french tourists الابحاث الاقتصادية/ economic research جنوب افريقيا بعد/ south africa after الحدودي مع مصر/ egypt/ border المخاوف المشروعة/ legitimate concerns هونغ كونغ ( احد/ ) hong kong العام 1961/1961 150 مليون شخص/ 150 million people في مدينة القدس/ in jerusalem الاسرائيلي موشيه كاتساف/ moshe katsav ضغط دولي/ international pressure جبل الي/ generation to/ جيانغ تسه من كرئيس/ jiang zemin as chairman دورية الشرطة/ police patrol الاليس 86 billion/86 alaves 33/33 في الغابون/ in gabon/ وسط تقارير/ amid reports/ بين برلين/ between berlin/ الخارجية الامريكية نيكلوس/ 7 nicholas iran supports/ ايران تساند/ dialoge with israel/ الحوار مع اسرائيل/ russia 5/5 روسيا north of basra/ شمال البصرة/ bulgaria 2/2 بلغاريا 7 15 8/8 15 الارجنطين ويران/ iran ملايين جنيه/ million pounds حتى الان من هذا/ so far this/ peter struck/ peter ان صرح/ said عناصر مع هذه/ members of the/ الاسرائيلي زيف/ zeev الافواج الفوري/ immediate release of/ اعلن متحدث باسم قوات التحالف/ coalition spokesman said/ with rwanda/ توافق بدون/ accept without/ اطباء فلسطينيون/ palestinian doctors/ اوروبية كثيرة/ many european / / هارتس/ ha 'aretz والسودان وسوازيلاند/ , swaziland sudan السبت مما/ saturday اكد شهود/ 1 witnesses 1.4 مليار يوان (//) billion yuan او بدونها/ or without مظفر اباد/ muzaffarabad مني مليار/ 200 billion على بعد حوالي/ kilometres الفحص والموافقة/ examination and approval/ on all fronts/ المصالحة والديمقراطية/ reconciliation and democracy موافقته المدينة/ agreed in principle/ سياسة الشمس المشروقة/ sunshine policy/ اللبناني فارس/ fares/ ليبيري على/ liberians to/ radar/ الرادار/ on the level/ الرئيس ديديه/ didier/ لكن بيريز/ peres/ جوزيب بيكه الذي/ josep pique , whose/ give details/ جزئية/ partial elections/ اجمالي 63/63 a total of 63/63 مزدحمة بالكاب/ crowded/ جوف هون/ geoff hoon/ الى 96/96 يعط تفاصيل/ give details/ remaining obstacles/ والخيرات في/ in/ and experience/ نوع اباتشي/ apache امن الدولة في/ s' state security/ تعاون سعودي/ saudi/ cooperation mutual/ ahmet/ احمد/ كان سابقا/ previously/ يقود الا/ only lead/ مستوى لها/ level/ كان موسى/ moussa/ مأرب مطالبين/ demanding/ maariib , mutual/ non-food/ الدعم المتبادل/ non-food/ مساجد ومستشفيات/ from haifa/ من حيفا/ eu and china/ الصين/ eu and china/ support - نيويورك -/: new york مصر لاجراء/ egypt for/ انه يتحدث باسم/ be speaking for/ اندرو سميث/ andrew smith/ تدخل دولية/ international intervention

Figure 7: 100 high probability Arabic-English lexical rules extracted by the two methods.

ITG (Saers et al., 2010). In addition, most of these works except for Levenberg et al. (2012) use small datasets with fewer than 100K sentence pairs.

More recently Sankaran et al. (2012a) proposed a Bayesian model employing Variational inference for directly extracting Hiero grammar from aligned phrase pairs. While our work is similar to theirs, there are notable differences as well. Firstly their model extracts a simpler arity-1 grammar, where the source and target sides can have at most 1 non-terminal. In contrast the grammar extracted by our model fully conforms to Hiero-style rules and hence can potentially capture larger reordering (than the unary grammar). Secondly, their inference does not scale for the larger datasets and consequently they train only on a subset of initial phrase-pairs thresholded by some high frequency. We further distribute the inference under a simple MapReduce style framework, which allows us to scale to large datasets.

A different line of work focuses on reducing the size of Hiero models, and we discussed these pa-

pers in Section 4.2. While this is not the central aspect of our work, we showed that simple pruning of the grammar (extracted by our model) could reduce the size by more than 80% without sacrificing translation quality.

## 6 Conclusion

This paper introduced a novel Bayesian model for learning Hiero SCFG translation rules which is an alternative to the commonly used heuristic rule extraction approach. For inference, we use Variational-EM along with a Map-Reduce style framework for distributing the training process. This allowed us to efficiently train the model for very large corpora. We provided quantitative results and also a detailed qualitative analysis to demonstrate the superiority of the model trained by our approach. In future work, we would like to extend our model for inference directly on full sentence pairs as has been applied for the syntax-based model (Galley et al., 2006).

## References

- Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. Bayesian synchronous grammar induction. In Proceedings of the Neural Information Processing Systems.
- Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In Proceedings of the Annual meeting of Association of Computational Linguistics.
- Rens Bod. 1998. Beyond Grammar: An Experience-Based Theory of Language. CSLI Publications, Stanford.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 263–270. Association for Computational Linguistics.
- David Chiang. 2007. Hierarchical phrase-based translation. Computational Linguistics, 33.
- Trevor Cohn and Gholamreza Haffari. 2013. An infinite hierarchical bayesian model of phrasal translation. In Proceedings of the Annual Meeting of Association for Computational Linguistics.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics.
- Zhongjun He, Yao Meng, and Hao Yu. 2009. Discarding monotone composed rule for hierarchical phrase-based statistical machine translation. In Proceedings of the 3rd International Universal Communication Symposium.
- Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Barga, and William Byrne. 2009. Rule filtering by pattern for efficient hierarchical translation. In Proceedings of the European Association for Computational Linguistics.
- Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.
- Abby Levenberg, Chris Dyer, and Phil Blunsom. 2012. A bayesian model for learning scfgs with discontinuous rules. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 223–232, Jeju Island, Korea. 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.
- Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. Computational Linguistics, 30:417–449.
- F. J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora, pages 20–28, University of Maryland, College Park, MD, USA.
- Markus Saers, Joakim Nivre, and Dekai Wu. 2010. Word alignment with stochastic bracketing linear inversion transduction grammar. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics.
- Baskaran Sankaran, Gholamreza Haffari, and Anoop Sarkar. 2011. Bayesian extraction of minimal scfg rules for hierarchical phrase-based translation. In Proceedings of the Sixth Workshop on SMT.
- Baskaran Sankaran, Gholamreza Haffari, and Anoop Sarkar. 2012a. Compact rule extraction for hierarchical phrase-based translation. In The 10th biennial conference of the Association for Machine Translation in the Americas (AMTA), San Diego, CA. Association for Computational Linguistics.
- Baskaran Sankaran, Majid Razmara, and Anoop Sarkar. 2012b. *Kriya* – an end-to-end hierarchical phrase-based mt system. The Prague Bulletin of Mathematical Linguistics (PBML), 97(97):83–98.
- Hendra Setiawan, Min-Yen Kan, Haizhou Li, and Philip Resnik. 2009. Topological ordering of function words in hierarchical phrase-based translation. 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 324–332. Association for Computational Linguistics.
- Jason Wolfe, Aria Haghighi, and Dan Klein. 2008. Fully distributed EM for very large datasets. In Proceedings of the 25th international conference on Machine learning. ACM.
- Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23(3):377–403.
- Mei Yang and Jing Zheng. 2009. Toward smaller, faster, and better hierarchical phrase-based smt. In Proceedings of the ACL-IJCNLP.
- Hao Zhang, Daniel Gildea, and David Chiang. 2008. Extracting synchronous grammar rules from word-level alignments in linear time. In Proceedings of the COLING.
- Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical mt. In Proceedings of the COLING.

# A Topic-Triggered Language Model for Statistical Machine Translation

Heng Yu<sup>†</sup>, Jinsong Su<sup>\*</sup>, Yajuan Lü<sup>‡</sup>, Qun Liu<sup>‡ †</sup>

<sup>†</sup>Institute of Computing Technology, Chinese Academy of Sciences

<sup>\*</sup> Software School of Xiamen University

<sup>‡</sup> Centre for Next Generation Localisation

Faculty of Engineering and Computing, Dublin City University

{yuheng, lvajuan, liuqun}@ict.ac.cn

jssu@xmu.edu.cn

qliu@computing.dcu.ie

## Abstract

Language model is an essential part in statistical machine translation, but traditional  $n$ -gram language models can only capture a limited local context in the translated sentence, thus lacking the global information for prediction. This paper describes a novel topic-triggered language model, which takes into account the topical context by estimating the  $n$ -gram probability under the given topics and online adjusts language model score according to different topic distributions. Experimental results show that our method provides a average improvement of +0.76 B on NIST Chinese-to-English translation task and a reduction in word perplexity of the test-document.

## 1 Introduction

Language model (LM) measures the fluency of translation outputs (Brown et al., 1993), and plays an important role in statistical machine translation (SMT). Traditional language model predicts the next word conditioning only on the preceding  $n-1$  words, thus ignores syntactic structures in the sentence and global information over the document.

One direct approach to handle this problem is to explore sentence-level context, such as syntax-based language model for reranking (Charniak et al., 2003), and dependency language model for String-to-Dependency model (Shen et al., 2008). But these methods are still not robust enough to handle the polysemy and domain changes, as they lack the global-context information.

Another interesting line is to utilize information at document-level. Intuitively, different domains or topics have different  $n$ -gram probability distributions. Thus, we should take into account the topic information when we translate a document. Topic model has been learned in several

parts of SMT, such as word-alignment (Zhao and Xing, 2006; Zhao and Xing, 2007; Gong et al., 2011), translation model (Xiao et al., 2012). All these works show that a particular translation often appears in some specific topical context, so it is reasonable to enhance the prediction ability of language model by incorporating topical information. Tan et al. (2011) introduces a large scale distributed composite language model incorporating document-level information. But they only focus on the target side and explore in  $n$ -best reranking task which has a limited search space, while another promising application is taking account of topical information on both sides and integrate the LM into decoding to online select translation hypotheses. However, the integration is not easy. Since the test-document can be from any topic, it is hard to dynamically estimate language model probability according to various topic distributions.

In this paper, we follow this line and introduce a novel topic-triggered language model. We first estimate the topic distribution for each document in training data, and assign those topic probabilities to each sentence. With target-side topic probabilities, we train a topic-specific language model for each topic. Then, rather than limiting topical context to target side, we utilize the source-side topical information at decoding time and online adjust language model score according to the topic distribution of the translated-document. As there is no explicit correspondence between topics on both sides, we project the source-side topic distribution to the target side as a trigger to our topic specific language models. As compared with previous works, our model takes advantage of the topical information on both sides, thus breaking down the context barrier for language model. Experimental results on various Chinese-English test sets show that our method gains an average improvement of +0.76 B points and a perplexity reduction over

the baseline model.

## 2 Related Work

Previous works devoted to improving language models in SMT mostly focus on utilizing more contextual information, such as syntax-based LMs (Charniak et al., 2003; Schwartz et al., 2011; Shen et al., 2008; Hassan et al., 2009), Forward & MI trigger LM (Xiong et al., 2011), and large-scale language models (Zhang et al., 2006; Brants et al., 2007; Emami et al., 2007; Talbot and Osborne, 2007). Since our philosophy is fundamentally different from them in that we incorporate information at document level to build language models. So we discuss previous works that explore topic information for SMT in this section.

Researchers have been trying to incorporate topic information into language models in several ways. Gildea and Hofmann (1999) use EM algorithm to perform a topic factor decomposition based on a segmented training corpus. They estimate unigram topic-based probability and combine it with standard  $n$ -gram model. Tam et al. (2007) and Ruiz and Federico (2011) introduce topic model for cross-lingual language model adaptation task. They use bilingual topic model to project latent topic distribution across languages. Based on the BLSA, they are able to transfer source-side topic weights into target-side and use them to generate topic-based marginals to adapt  $n$ -gram language model. Our model is different from theirs in that rather than using topic-based probabilities to adapt  $n$ -gram model, we directly calculate LM probability conditioned on topic distributions.

There are also some valuable applications of topic model for machine translation. Zhao and Xing (2006) propose the Bilingual Topic Admixture Model (BiTAM) for word alignment and extract topic-dependent translation model accordingly. Gong et al. (2011) introduce topic model for filtering topic-mismatched phrase pairs. Su et al. (2012) use the topic distribution of in-domain monolingual corpus to adapt the translation model. Xiao et al. (2012) introduce a topic similarity model to select the synchronous rules for hierarchical phrase-based translation. Our work is in the same spirit with those works, but we are interested in LM problem rather than other parts in SMT.

Our work models topic probabilities into training corpus and trains several topic-specific LMs,

so it is in the same spirit of mixture modeling. Heidel et al. (2007) use topic distribution to cluster the training corpora and train LMs accordingly. Our method is different from theirs in that we assign topic probabilities to training sentences rather than segment them into different topics, so our model is more robust to data sparse problem. Besides, Foster and Kuhn (2007), Civera and Juan (2007), Lü et al. (2007) also adapt mixture modeling framework to exploit the full potential of existing corpus. Adopting this framework, the training corpus is first divided into different parts, each of which is used to train a sub model, then these sub models are used together with different weights during decoding. Those works typically use word similarities and sentence level information, while our work extends the context into the document level.

## 3 Topic triggered Language Model

Polysemy is a difficult problem for statistical machine translation. As shown in Figure 1, English sentence "give me a shot" has different meanings in different domains. Using traditional LM, which only considers the local context information in the translated sentence, this ambiguous translation is hard to handle, since these translations are all common in the corpus with different domains. But with the help of topical context information, the difference can be told. For example, the word 'shot' is often translated into "(photo)" in the sentences related to the film topic, and to "(chance)" in sports topic. So as the topic information is concerned, LM allows for more fine-grained distinction of different translations and enjoys stronger prediction power.

In our method, we introduce the topic of current document  $t$  as a hidden variable, and decompose the language model probability as follows:

$$P(\mathbf{e}) = \sum_t P(\mathbf{e}, t) = \sum_t P(\mathbf{e}|t) \cdot P(t) \quad (1)$$

$P(\mathbf{e}|t)$  indicates the probability of the sequence  $\mathbf{e}$  given the topic  $t$ , and  $P(t)$  is the topic distribution of the test-document which is calculated during decoding. In general, our framework to build the topic-trigger language model can be specified into two steps:

- Build topic-specific LMs conditioned on the topic distribution estimated by the target-side topic model

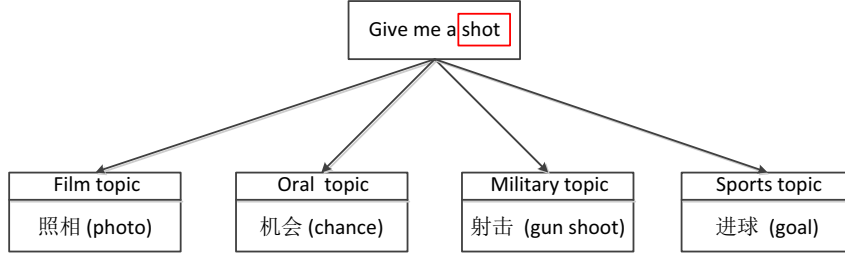


Figure 1: Example of different translations of word "shot" in different topics

- Capture source-side topic information during decoding and online adjust LM score

We will give detailed description of the two parts in the following section.

### 3.1 Topic-specific language model

In this section, we first briefly review the principle of Hidden Topic Markov Model (HTMM) which is the basis of our method, then describe our approach to build topic-specific LMs in detail.

#### 3.1.1 Hidden Topic Markov Model

Topic model is a suite of algorithms aiming to discover the hidden thematic structure in large archives of documents. Recently both Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) have been successfully applied in various NLP tasks. Based on the "bag-of-words" assumption that the order of words can be ignored, these methods model the corpus as a co-occurrence matrix of words and documents, and build generative models to infer the latent aspect of topics. Using these models, words can be clustered into the derived topics with a probability distribution. and the correlation between words can be automatically captured via topics.

However, the "bag-of-words" assumption is an unrealistic oversimplification in language model case because it ignores the order of words which is critical in estimating  $n$ -gram probabilities. To remedy this problem, we use Hidden Topic Markov Models (HTMM), proposed by (Gruber et al., 2007), which models the topics of words in the document as a Markov chain. The model is based on the assumption that all words in the same sentence share the same topic and the successive sentences are more likely to have the same topic. HTMM incorporates the local dependency between words by Hidden Markov Model for better topic estimation.

#### 3.1.2 Topic Probability Assignment

We use HTMM (Gruber et al., 2007) to train topic model on our training set and obtain sentence-level topic probabilities. To avoid data sparse problems, we use the topic probability of each sentence as a soft clustering for each topic rather than force hard decisions on topic assignment. In this way, we are able to get  $n$ -gram distributions for different topics. So the topic-sensitive words will have a higher occurrence in specific topics while common words will distribute uniformly in every topic.

#### 3.1.3 Estimation

We follow the common practise in  $n$ -gram model (Goodman, 2001) and simplify  $P(\mathbf{e}|t)$  into a serial of  $n$ -gram probabilities  $P(w_i|w_{i-n+1}^i, t)$  based on Markov Assumption. Formally, we decompose the probability as follows:

$$P(\mathbf{e}|t) = P(w_1|t) \cdot P(w_2|w_1, t) \cdots P(w_i|w_{i-n+1}^i, t) \quad (2)$$

Noted that, based on HTMM, we assume that all words in one sentence share the same topic, so topic  $t$  in Equation 2 can be shared. To compute  $P(w_i|w_{i-n+1}^i, t)$ , We use Maximum-Likelihood Estimation (MLE) with the  $n$ -gram fractional count for each topic. And since some topic-based  $n$ -grams probabilities are sharply distributed, we use Witten-Bell(WB) method (Witten and Bell, 1991) for smoothing.

$$P_{MLE}(w_i|w_{i-n+1}^{i-1}, t_e) = \frac{Count(w_i|w_{i-n+1}^{i-1}, t_e)}{Count(w_{i-n+1}^{i-1}, t_e)} \quad (3)$$

$$P(w_i|w_{i-n+1}^{i-1}, t_e) = \lambda_{w_{i-n+1}^{i-1}} P_{MLE}(w_i|w_{i-n+1}^{i-1}, t_e) + (1 - \lambda_{w_{i-n+1}^{i-1}}) P(w_i|w_{i-n+2}^{i-1}, t_e) \quad (4)$$

In Equation 4,  $\lambda$  is a normalization parameter for MLE probability and back-off probability,

which can be calculated using the following equation:

$$\lambda_{w_{i-n+1}^{i-1}} = \frac{N_{1+}(w_{i-n-1}^{i-1}, t_e)}{N_{1+}(w_{i-n-1}^{i-1}, t_e) + \sum_{w_i} c(w_{i-n+1}^i, t_e)} \quad (5)$$

where  $N_{1+}(w_{i-n-1}^{i-1}, t_e)$  denotes for the number of words  $w$  following  $w_{i-n-1}^{i-1}$  in topic  $t_e$ , and  $c(w_{i-n+1}^i, t_e)$  is the count of  $n$ -gram  $w_{i-n+1}^i$  in  $t_e$ .

### 3.2 Integration with SMT

We integrate our LM into SMT system to utilize topic distribution of the test-document as a trigger to each topic-specific language model. But as we know, only source side is available before decoding in SMT. So in order to get target-side topic distribution  $P(t_e)$ , we need to estimate the source-side topic distribution  $P(t_f)$  and then project it to the target side. So Equation 1 can be further refined as the following Equation:

$$P(\mathbf{e}) = \sum_{t_e} P(\mathbf{e}|t_e) \cdot \sum_{t_f} P(t_e|t_f) \cdot P(t_f) \quad (6)$$

where  $P(t_e|t_f)$  is the topic projection probability.

#### 3.2.1 Topic Projection

Since topic distributions of bilingual sentences often share the same pattern (Gao et al., 2011), we follow the work of Xiao et al. (2012) and introduce the topic projection probability  $P(t_e|t_f)$  to project the source-side topic distribution into the target-side topic space. We train topic models on both sides of the training data, then with the help of the word alignment we estimate the projection probability by the co-occurrence of the source-side and the target-side topic assignment.

Formally, we denote each parallel sentence pair by  $(t_f, t_e, \mathbf{a})$ , where  $t_f$  and  $t_e$  are the topic assignments of source-side and target-side sentences respectively, and  $\mathbf{a}$  is a set of word alignments  $\{(f_i, e_j)\}$ . An alignment  $(i, j)$  denotes source-side word  $f_i$  aligns to target-side word  $e_j$ , so the topics of both words are also aligned. Thus, the co-occurrence of a source-side topic with index  $d_f$  and a target-side topic  $d_e$ ,  $Cnt(t_f, t_e)$  is calculated by:

$$Cnt(t_f, t_e) = \sum_{(t_f, t_e, \mathbf{a})} \sum_{(i, j) \in \mathbf{a}} \delta(t_{f_i}, d_f) * \delta(t_{e_j}, d_e) \quad (7)$$

where  $\delta(x; y)$  is the Kronecker function, which is 1 if  $x = y$  and 0 otherwise. We then compute the

probability of  $P(t = d_f, t = d_e)$  by normalizing the co-occurrence count. Overall, we obtain a correspondence matrix  $M_{d_e \times d_f}$  from target-side topic to source-side topic, where the item  $M_{i,j}$  represents the probability  $P(t_f = i, t_e = j)$ . Then with the correspondence matrix  $M_{d_e \times d_f}$ , we are able to project the source-side topic  $P(t_f)$  to the target-side topic space, which we called projected target-side topic distribution  $T(P(t_f))$ .

#### 3.2.2 Topic-triggered Estimation

During decoding, we first estimate the source-side topic distribution of the test-set  $P(t_f)$ , then using the topic projection matrix, we map  $P(t_f)$  to the target side, and generate each topic  $t_e$  with probability  $P(t_e|t_f)$ . Then topic  $t_e$  triggers its topic-specific LM  $P(e|t_e)$ . We use the weighted sum of each model as the final LM score.

## 4 Experiments and Results

We try to answer the following questions by experiments:

- Can our topic-triggered language model help improve translation quality in terms of both B and perplexity.
- How is the topic number affect the language model performance.
- Can our model make better use of training corpus than N-gram model.

### 4.1 Experiment setup

We present our experiments on the NIST Chinese-English translation tasks. The bilingual training data for translation model contain 1.5M sentence pair with 38M Chinese words and 32M English words. The monolingual data for training English language model includes the Xinhua portion of the GIGAWORD corpus, which contains 10M sentences. We used the NIST evaluation set of 2006(MT06) as our development set, and sets of MT04/05/08 as test sets. Corpus statistics are shown in Table 1.

We obtain symmetric word alignments of training data by first running GIZA++ (Och and Ney, 2004) in both directions and then applying refinement rule "grow-diag-final-and" (Koehn et al., 2003). We re-implement the Hierarchical phrase-based system (Chiang, 2007) and extract SCFG

Data	Sentence	documents
Language model training	10M	980K
Translation model training	1.5M	99.4K
Tuning	616	52
Testing(04)	1788	200
Testing(05)	1082	100
Testing(08)	1357	109

Table 1: Training, tuning and test data used for evaluating B score.

rules from this word-aligned training data. A 4-gram language model is trained on the monolingual data by SRILM toolkit (Stolcke, 2002). Case-insensitive NIST BLEU (Papineni et al., 2002) is used to measure translation performance. We use minimum error rate training (Och, 2003) for optimizing the feature weights.

To obtain topic distribution, We use the open source LDA tool Open HTMM developed by Gruber et al. (2007) for estimation and inference. During this process, we empirically set the parameter values for HTMM training as:  $\alpha = 1.5, \beta = 1.01, iters = 100$ . See Gruber et al. (2007) for the meanings of these parameters. and set the topic number to  $30^1$  for both source and target side. The source-side topic model is estimated from the Chinese part of training corpus, while the target side is estimated from both Xinhua and the English side of training corpus.

#### 4.2 Effect of topic-trigger language model

For machine translation task, our baseline is the traditional hiero system with standard features (Chiang, 2007). The baseline language model is a 4-gram model trained on Xinhua corpus. Noted that we use Keneser-Ney smoothing (Kneser and Ney, 1995) for baseline LM since it’s universally acknowledged to achieve better performance. And our topic-triggered language model is trained on the same corpus with topic distribution estimated from topic model. We add our model as a new feature into the system, denote as STLM. To prove the soundness of our approach, we re-implement two comparative experiments: HTLM makes hard-decision on topic selection in both training and decoding, assigning the topic with the highest probability to the sentence, which is in the same spirit with the Heidel et al. (2007) method. Second, we

<sup>1</sup>We determine the topic number by testing 5, 10, 15, 30, 50 in our preliminary experiments. We find that 30 topics produces a slightly better performance than other values.

ppl	04	05	08
Base LM	158.42	134.59	208.11
Topic LM	148.11	119.17	200.41

Table 3: 4-gram word perplexity results of our method in terms of *ppl*. We compare our model with baseline *n*-gram model (“Base LM”) on three test-sets.

follow the method by Tam et al. (2007), denote as “Tam”, and generate topic-based marginals to adapt *n*-gram language model.

Table 2 reports the B and TER scores on all test-sets. The baseline system achieves B score of 37.43 on NIST04, 33.67 on NIST05 and 28.54 on NIST08 set. Our method(STLM) gains an average improvement of +0.76 B and an average reduction of -0.88 TER over the baseline. Results on NIST MT 04, 05, 08 are statistically significant with  $p < 0.05$  (Koehn, 2004). This verifies that our topic-triggered language model is a good complement for *n*-gram model to further improve translation quality. We can also see that our method generally out-performs the Tam’s method, because our model can capture *n*-gram level topic information, rather than only focus on estimating 1-gram topic-based probability. Another interesting result is forcing hard-decision on topic selection (HTLM) only achieves a little improvement over the baseline. The reason is two folded: First, in LM training process, the hard-decision on topic will serve as a corpus split strategy and cause data sparse problems. Second in decoding, one sentence may not solely belong to one topic, so the hard decision will cause inaccurateness in LM prediction.

We then evaluate our method in terms of perplexity. As an initial measure to compare language models, average per-word perplexity(*ppl*), reports how surprised a model is by test data. Equation 8 calculates *ppl* using log base *b* for a test set of *T* tokens.

$$ppl = b^{\frac{-\log_b P(e_1 \dots e_T)}{T}} \quad (8)$$

we evaluate 4-gram perplexity of the translation hypotheses using baseline language model and our topic-triggered model.

Table 3 shows that our model reduces the average word perplexity by 6% compared to baseline language model. The results indicate that our model successfully leverages the source-side document and reduces the *ppl* on the target side.



Model	04		05		08		AVG	
	B	TER	B	TER	B	TER	B	TER
Baseline	37.43	39.88	33.67	42.45	28.54	47.32	33.21	43.22
Tam	37.86	39.12	34.28	41.93	29.02	46.92	33.72	42.66
HTLM	37.46	39.86	33.74	42.41	28.67	47.23	33.29	43.17
STLM	<b>38.28</b>	<b>38.95</b>	<b>34.30</b>	<b>41.93</b>	<b>29.32</b>	<b>46.14</b>	33.97	42.34

Table 2: Results of our method in terms of B /TER, "Tam" denotes using topic adaptation method from Tam et al. (2007). "HTLM" denotes using topic-triggered LM with hard decision of topic assignment, and "STLM" means topic assignment by probabilities. Scores marked in bold are statistically significantly with  $p < 0.05$  (Koehn, 2004).

Test-set	04	05	08
baseline	31.31	28.43	23.67
5 topics	30.81	27.96	23.26
10 topics	30.98	28.12	23.42
15 topics	31.32	28.40	23.64
20 topics	31.39	28.40	23.82
30 topics	31.75	28.51	24.05
50 topics	31.70	28.48	24.01

Table 4: Results on all test sets with different topic number.

### 4.3 Effect of topic number

In topic model training, topic number is a manually set parameter. However, as an empirical factor, the topic number diverse a lot in different training corpus. so it's worthy to explore the effect of topic number on the performance of our topic-trigger language model. We set topic number to 5, 10, 15, 20, 30, 50 respectively to train topic models on both sides.

Table 4 shows the B scores using 5, 10, 15, 20, 30, 50 topics. We can see that with only 5 topics, the model performance is a little worse than the baseline model. This is reasonable because the corpus has not been fully clustered into different topics, so the topic information has not been fully utilized. But we can see ,as the topic number grows larger, the performance gets better with a peak at 30 topics, resulting a 0.34 improvement average over the baseline.

But there is a little slump when it comes to 50, we think the reason is as we models topic distribution into the LM training corpus, the distribution gets too scattered as topic number grows causing data-sparse problem in topic-specific language model training, thus affecting the overall probability of the language model.

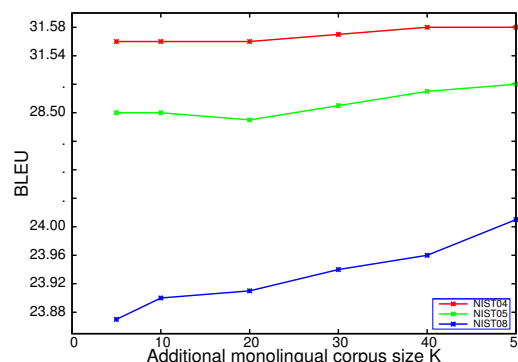


Figure 2: BLEU improvement with additional topic-modeling training corpus

### 4.4 Effect of better topic model estimation

Finally, we investigate the effect of larger topic-training corpus. One important feature of topic model is the larger the training corpus is , the better model we will get. In our experiment, we use the source of fbis which only have 10,947 documents to train source-side topic model. This may not be good enough to correctly estimate the topic distribution of the test set, since we know that NIST08 contains a large portion of web corpus. So we add different size of source-side monolingual corpus: 5K, 10K, 20K, 30K, 40K, 50K from Chinese Sohu weblog corpus<sup>2</sup> only to train different source-side topic models with 30 topics.

Figure 4 shows the B scores of the translation system on NIST04,05,08. It can be seen that additional corpus improves translations quality on NIST08. This is because the additional corpus expand the diversity of the topic model, especially for NIST08 which contains a large part of web data, generating more accurate topic distribution. The best B comes to 24.12 when the additional corpus size is 50K, achieving 0.42 gains on the

<sup>2</sup><http://blog.sohu.com>

baseline system. But on 04 and 05 test-sets, the improvement is not that significant. This may be because the 04, 05 set are not similar with the additional corpus, so they are not effected by the improvement of topic model. The results indicates that the performance of our topic-triggered language model is directly associated with the topic model, which can be improved by training with larger and more relative corpus.

## 5 Conclusion

In this paper, we follow this line and introduce a novel topic-triggered LM. We first estimate the topic distribution for each document in training data, and assign those topic probabilities to each sentence, then, we train a topic-specific  $n$ -gram LM for each topic based on those topic probabilities. At decoding time, as target translations are not available before translation, we simply project the topic distribution from source to target side. Then we compute the topic-triggered LM score according to the topic distribution of the translated-document. Experimental results show that our model achieves better performance than traditional  $n$ -gram model on both perplexity and B score.

In the future, we will verify our method in other domain and language pairs. Further more, we want to combine our work with other related works to see if it can further improve the translation quality. Finally, we will explore more robust framework to incorporate syntax and semantic information to make our language model more powerful.

## 6 Acknowledgments

The authors were supported by 863 State Key Project (No. 2011AA01A207), and National Key Technology R&D Program (No. 2012BAH39B03). Jinsong Su's work is supported by Reseach Fund for the Doctoral Program of Higher Education of China (Grant NO. 20120121120046). Qun Liu's work is partially supported by Science Foundation Ireland (Grant No.07/CE/I1142) as part of the CNGL at Dublin City University. We would like to thank the anonymous reviewers for their insightful comments and those who helped to modify the paper.

## References

- David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent dirichlet allocation. In *Journal of Machine Learning Research*, volume 3, page 2003.
- Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, Jeffrey Dean, and Google Inc. 2007. Large language models in machine translation. In *EMNLP*, pages 858–867.
- Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311, June.
- Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *In MT Summit IX. Intl. Assoc. for Machine Translation*.
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.
- Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation, StatMT '07*, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- A. Emami, K. Papineni, and J. Sorensen. 2007. Large-scale distributed language modeling. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–37 –IV–40, april.
- George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation, StatMT '07*, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. 2011. Clickthrough-based latent semantic models for web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11*, pages 675–684, New York, NY, USA. ACM.
- Daniel Gildea and Thomas Hofmann. 1999. Topic-based language models using em. In *In Proceedings of EUROASPEECH*, pages 2167–2170.
- Z. Gong, G. Zhou, and L. Li. 2011. Improve smt with source-side "topic-document" distributions. In *Machine Translation Summit XIII*, page 496.
- Joshua T. Goodman. 2001. A bit of progress in language modeling. In *Technical Report MSR-TR-2001-72*.
- Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. 2007. Hidden topic markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 163–170.

- Hany Hassan, Khalil Sima'an, and Andy Way. 2009. A syntactified direct translation model with linear-time decoding. In *Proceedings of EMNLP 2009*, pages 1182–1191, Singapore, August. Association for Computational Linguistics.
- Aaron Heide, Hung an Chang, and Lin-Shan Lee. 2007. Language model adaptation using latent dirichlet allocation and an efficient topic inference algorithm. pages 2361–2364.
- T. Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.
- R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181 – 184 vol.1, may.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL 2003*, pages 127–133.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP 2004*, pages 388–395.
- Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *EMNLP-CoNLL*, pages 343–350.
- Franz Joseph Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, pages 417–449.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL 2003*, pages 160–167.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL 2002*, pages 311–318.
- Nick Ruiz and Marcello Federico. 2011. Topic adaptation for lecture translation through bilingual latent semantic models. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT '11*, pages 294–302, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lane Schwartz, Chris Callison-Burch, William Schuler, and Stephen Wu. 2011. Incremental syntactic language models for phrase-based translation. In *Proceedings of ACL 2011*, pages 620–631, June.
- Libin Shen, Jinxi Xu, and Ralph M Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *ACL*, pages 577–585.
- Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proc. of ICSLP 2002*, pages 901–904.
- Jinsong Su, Hua Wu, Haifeng Wang, Yidong Chen, Xiaodong Shi, Huailin Dong, and Qun Liu. 2012. Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of ACL 2012*, pages 459–468, Jeju Island, Korea, July. Association for Computational Linguistics.
- David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Prague, Czech Republic. Association for Computational Linguistics*, pages 512–519.
- Yik-Cheung Tam, Ian Lane, and Tanja Schultz. 2007. Bilingual-lsa based lm adaptation for spoken language translation. In *Proceedings of ACL 2007*, pages 520–527, Prague, Czech Republic, June. Association for Computational Linguistics.
- Ming Tan, Wenli Zhou, Lei Zheng, and Shaojun Wang. 2011. A large scale distributed syntactic, semantic and lexical language model for machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 201–210, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Ian H Witten and Timothy C Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, 37(4):1085–1094.
- Xinyan Xiao, Deyi Xiong, Min Zhang, Qun Liu, and Shouxun Lin. 2012. A topic similarity model for hierarchical phrase-based translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 750–758, Jeju Island, Korea, July. Association for Computational Linguistics.
- Deyi Xiong, Min Zhang, and Haizhou Li. 2011. Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. In *ACL*, pages 1288–1297.
- Ying Zhang, Almut Silja, and Hildebrand Stephan Vogel. 2006. Distributed language modeling for n-best list re-ranking. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.
- B. Zhao and E.P. Xing. 2006. Bitam: Bilingual topic admixture models for word alignment. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 969–976. Association for Computational Linguistics.
- Bing Zhao and Eric P Xing. 2007. Hm-bitam: Bilingual topic exploration, word alignment, and translation. In *Advances in Neural Information Processing Systems*, pages 1689–1696.

# Reserved Self-training: A Semi-supervised Sentiment Classification Method for Chinese Microblogs

Zhiguang Liu Xishuang Dong Yi Guan\* Jinfeng Yang

School of Computer Science and Technology

Harbin Institute of Technology, Harbin, China

{triggerliu, dongxishuang, yangjinfeng2010}@gmail.com  
guanyi@hit.edu.cn

## Abstract

The imbalanced sentiment distribution of microblogs induces bad performance of binary classifiers on the minority class. To address this problem, we present a semi-supervised method for sentiment classification of Chinese microblogs. This method is similar to self-training, except that, a set of labeled samples is reserved for a confidence scores computing process through which samples that are less than a predefined confidence score threshold are incorporated into training set for retraining. By doing this, the classifier is able to boost the performance on the minority class samples. Experiments on the NLP&CC2012 Chinese microblog evaluation data set demonstrated that reserved self-training outperforms the best run by 2.06% macro-averaged and 2.30% micro-averaged F-measure, respectively.

## 1 Introduction

Sentiment classification aims to label peoples opinions as different categories such as positive and negative from a given piece of text (Pang et al., 2002). Currently, related research on traditional online media, such as blogs, forums, and online reviews, has made great progress (Banerjee and Agarwal, 2012; Liu et al., 2005). However, sentiment classification of microblogs is hard to process due to some unique characteristics of microblogs, for example, short length of update messages and language variations. Moreover, topic based microblogs are related with peoples daily lives and people are more likely to post some negative messages to show their unsatisfactoriness, which may partially result in imbalanced sentiment class distributions. For example, the num-

ber of negative tweets is far more than that of positive in some topics, which is different from the previous work on sentiment classification that assumes the balance between positive and negative samples (Chawla et al., 2002; Yen and Lee, 2009).

While supervised techniques have been widely used in sentiment classification (Pang et al., 2002), the main problem that supervised methods suffered is that they rely on labeled data solely. Semi-supervised methods, which make use of both labeled and unlabeled data, are ideal for sentiment classification, since the cost of labeling data is high whereas unlabeled data are often readily available or easily obtained (Ortigosa-Hernández et al., 2012). However, there are some drawbacks of semi-supervised approaches such as most of the work assume that the positive and negative samples in both labeled and unlabeled data set are balanced, otherwise models often bias towards the majority class (Chawla et al., 2002; Yen and Lee, 2009). In addition, most existing studies on imbalanced classification focus on supervised learning methods, with few on semi-supervised approaches (Li et al., 2011).

In this study, we propose a reserved self-training method for binary sentiment classification inspired by active learning strategies (Ryan, 2011). Active learning systems interact with domain experts who are responsible for annotating unlabeled samples, and aim to achieve better performance with less training data (Wu and Ostendorf, 2013). The key to active learning is to find an appropriate query strategy such as the classifier poses queries to decide which samples are most informative. We randomly reserved a portion of labeled samples before training. Reserved self-training is the process of simulating active learning that repeatedly queries our reserved samples and then incorporates the labeled samples about which the classifier is least certain into training corpus for retraining, thus the retrained classifier is able to improve

\*Corresponding author: guanyi@hit.edu.cn

the performance of classification on the minority class.

The remainder of this paper is organized as follows. The next section reviews some related work on semi-supervised sentiment classification as well as imbalanced classification briefly. We formally define the task in section 3. Section 4 presents our approach of reserved self-training algorithm for imbalanced sentiment classification. Section 5 provides experimental results on a data set of 20 topics. Finally, section 6 summarizes the work, draws some conclusions, and suggests related future work.

## 2 Related Work

Sentiment classification ranges from the document level, to the sentence and phrase level, and we concentrate on sentence level classification. Sentence level sentiment classification methods can be categorized into three types: supervised (Pang et al., 2002) unsupervised (Turney, 2002), and semi-supervised learning methods (Singh et al., 2008) among which semi-supervised approaches are more appropriate for sentiment classification of microblogs due to their capability of making use of both labeled and unlabeled data. Another related work is imbalanced classification stems from several unique characteristics possessed by microblogs (A detailed study can be found in section 3.2).

### 2.1 Related Semi-supervised Sentiment Classification Works

Semi-supervised learning approaches make good use of a small portion of labeled and a large amount of unlabeled data to build a better classifier. One of the bottlenecks in applying supervised learning is that it needs to label many samples by domain experts. To save the work of manual annotation, Riloff et al. (2003) introduced a bootstrapping method which was able to automatically label training samples. They started on a few seeds for training, subsequently, incorporated five highest scores unlabeled samples into training corpus to retrain the model iteratively. Chang et al. (2007) added some restrictions to self-training, making it possible to produce better feedback information in the learning process. For a given classification task, one of the problems of adopting co-training is that it assumes two conditionally independent feature sets could be extracted (Blum and

Mitchell, 1998). Although further studies loosed this strong assumption (Balcan et al., 2004), two classifiers must be different enough to achieve complementation. Li et al. (2011) proposed a random subspace generation algorithm for co-training applied to imbalanced sentiment classification, but its corpus limited to English product reviews.

### 2.2 Related Imbalanced Classification Works

Imbalanced classification, as an appealing task, has been extensively studied in many research areas such as pattern recognition (Barandela et al., 2003) and data mining (Chawla et al., 2004). We pay special attention to resampling and cost-sensitive methods, since they are widely applied in imbalanced classification. Other methods such as induction technique and boosting (Weiss, 2004) are beyond the scope of this paper.

Resampling is a process in which the size of training samples is changed to modify the overall size and distribution of a corpus, among these methods downsampling and oversampling are two widely used resampling techniques. Downsampling (Barandela et al., 2003) takes a subset of majority classes samples whereas oversampling (Chawla et al., 2002) randomly repeats minority classes samples to keep balance between different classes. Downsampling needs shorter training time, at the expense of disregarding potentially useful samples. Oversampling increases the size of training data set that leads to a longer training time. Moreover, oversampling may cause over fitting due to minority class samples are randomly duplicated (Chawla et al., 2002; Drummond et al., 2003). In addition to the basic downsampling and oversampling techniques, there are some other sampling methods working in a more complicated fashion. SMOTE (Chawla et al., 2002) created some synthetic minority class examples and then performed a combination of oversampling and downsampling, which achieved better performance than only applying downsampling. Some other methods integrated different sampling strategies to obtain further improvement (Batista et al., 2004).

Cost-sensitive learning (Ling et al., 2004; Zadrozny et al., 2003) is another type of method used for dealing with imbalanced classification. Most cost-sensitive learning methods can be generally divided into two categories (Lee et al., 2012): transforming an existing cost-insensitive

classifier into an equivalent cost-sensitive via a wrapper approach, or taking the cost of misclassification into consideration when training a classifier by labeled samples.

### 3 Task Definition

We first give a formal definition of our task, and then analyze the unique characters of Chinese microblogs compared to traditional online media, such as forums and blogs.

#### 3.1 The Task

Our study involves classifying opinions of Chinese microblogs as either positive or negative. We perform sentence level sentiment classification for a given message of microblogs. We first conduct some preprocessing such as word segmentation and noisy symbols filtering. Subsequently, features for the classifier are extracted from each message. Finally, reserved self-training is employed to predict unlabeled data. Although we restrict the scope of study on Chinese microblogs, the method proposed in this study can be straightly extended in support of other languages such as English.

Here is an instance of illustrating our task. For a message “#50个人生必去的胜地# 万里长城太棒了，将来一定去看看！@李雷” (#The 50 places you must see# The Great Wall of China is amazing, I will visit it someday in future@ Lei Li, a Chinese name). The words between the # symbol refers to a relevant topic and the symbol @ means a mention or reply. This message is expected to be parsed into a triple: (Topic: The 50 places you must see), (Content: The Great Wall of China is amazing, I will visit it someday in future), (Polarity: Positive). Here, ‘Topic’ is a key word people interested; ‘Content’ refers to the content of a posted message; ‘Polarity’ denotes the predicted polarity produced by our model, and the possible values for it could be positive and negative.

#### 3.2 Characteristics of Sentiment Classification of Chinese Microblogs

Compared with traditional media such as blog and product reviews, detecting sentiment from microblogs is much harder due to the following challenges posed by microblogs. First, different from English, each written Chinese sentence need to be split into a sequence of words, however, the frequent use of informal and irregular words in mi-

croblogs may hinder the accuracy of segmentation. Second, the short length of messages and language variation contribute to the data sparsity problem. Third, different from previous work which concentrated on specific domain such as digital product reviews, sentiment classification of microblogs involves multi-domain information, thus, the model trained on one domain may perform badly when shift to another one. Lastly, the dynamic updates feature of microblogs means that sentiment class distributions may vary over time, in which case we need to handle imbalanced sentiment classification.

The NLP&CC2012<sup>1</sup> evaluation data set consists of 20 topics collected from Tencent Microblog<sup>2</sup>, involving multiple domains such as political, environmental, and health issues. Illustrated in Figure 1, it can be observed that all the classes of training corpus are biased. In particular, positive sentences account for the majority class in topic 3, 6, and 11, which is different from the other topics.

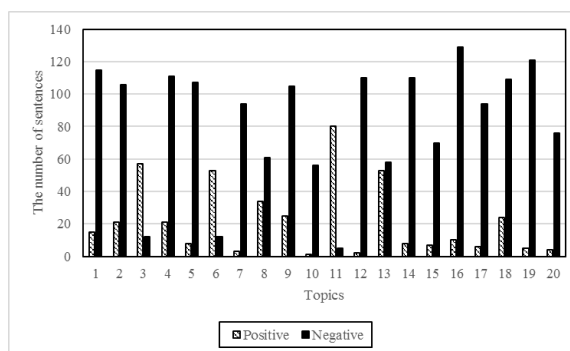


Figure 1: Class distributions of positive and negative samples in 20 topics

### 4 Reserved Self-Training for Imbalanced Classification

In this study, we incorporate a learning strategy into self-training, inspired by active learning, to tackle the imbalanced binary classification problems.

#### 4.1 Self-training

Self-training is a common method of semi-supervised learning which makes use of both the labeled and unlabeled data as training corpus. As shown in Algorithm 1, Self-training is a wrapper algorithm that iteratively applies supervised

<sup>1</sup><http://tcci.ccf.org.cn/conference/2012>

<sup>2</sup><http://t.qq.com>

---

**Algorithm 1** The self-training algorithm

---

**Input:**

Labeled data  $L$   
Unlabeled data  $U$ .

**Procedure:**

1. Apply supervised method to train a classifier  $C$  with  $L$ .
2. Make predictions on unlabeled data  $U$  with  $C$ .
3. Incorporate the most confidently predicted unlabeled data  $M$  in  $U$  along with each predicted label into  $L = L \cup M$ .
4. Loop for  $S$  iterations.

**Output:**

New labeled sample set  $L$  and classifier  $C$ .

---

method inside. It starts training on labeled data only, after each iteration, the most confidently predicted unlabeled samples would be incorporated as additional labeled data, decided by confidence scores calculation function. However, applying self-training to sentiment classification of Chinese microblogs in both subjectivity detection and sentiment classification performed not as well as expected, and the prediction results often bias towards the majority class. Comparing with fully supervised methods, the performance of self-training is even worse especially on the minority class (A detailed comparative study can be found in Section 5.2). It is not economical to revise the supervised classifier inside self-training, however, we may improve the data selection strategy to boost the performance of self-training on the minority class samples. Reserved self-training is such a technique that applies selection strategy in both labeled and unlabeled data during the learning process.

## 4.2 Reserved Self-training Classification

### Algorithm

In some cases, it seems unclear what self-training is really doing, and which theory it corresponds to (Chapelle et al., 2006). Intuitively, it is almost definite to label high confidence samples, namely with little effect on the model. However, the discriminative ability of the model could be significantly improved if we try to label those samples about which the classifier is least certain. Similar to self-training, the idea behind reserved self-training is quite simple except that we first re-

---

**Algorithm 2** The algorithm of reserved self-training for imbalanced sentiment classification

---

**Input:**

Labeled data  $L$  consisting of positive examples  $P$  and negative examples  $N$ , where  $|N| > |P|$ .

Unlabeled data  $U$  that is also imbalanced.

**Procedure:**

## Initialization:

1. Reserve a random portion  $R$  of  $L$ , and the remaining set  $L' = L - R$  is used for training.
2. Loop for  $M$  iterations
3. Train the classifier  $C$  with  $L'$ .
4. Make predictions on unlabeled data  $U$  with  $C$ .
5. Predict the reserved portion of labeled data  $R$  by classifier  $C$ .
6. Incorporate the most confidently predicted unlabeled data in  $U$  along with each predicted label into  $L'$ .
7. Incorporate the least confidently predicted labeled samples in  $R$  into  $L'$ .

**Output:**

New labeled sample set  $L'$  and classifier  $C$ .

---

serve a portion  $R$  of the training set  $L$  before training the initial classifier. As depicted in Algorithm 2, we apply the classifier to predict the unlabeled data  $U$  and the reserved data  $R$ , then we add those most confident unlabeled data and those least confident reserved data into training set  $T'$ . By adding training samples in this way, the classifier could increase the coverage of its decision space while not adding too many majority class samples. We use training set  $T'$  to train the model  $C$  iteratively until stopping criterion is met. Finally, assessing the performance of classifier  $C'$  on a labeled data set.

## 4.3 Labeled Data Selection

Generally, semi-supervised sentiment classification takes much less training data than supervised approaches, which forcing us to select the most effective samples from labeled data available. We resort to the principle of maximizing the diversity of samples in feature space to select seed. First, choose several samples as initial seed at random. Second, compute the centroid of the seed in feature space. Lastly, select those samples with least similarity to centroid of the seed done by cosine

similarity. By choosing seed in this manner, we aim to build a diversified data set to cover the feature space properly.

#### 4.4 Confidence Scores Calculation

For binary classification, we employ probabilistic model to determine the confidence to which class a given sentence belongs, in that case the classifier queries the samples whose posterior probability of being positive or negative is nearest to pre-defined threshold. In this study, we employ MaxEnt and SVM as basic polarity classification. Normally, we could obtain the predicted label along with their confidence scores by MaxEnt. SVM adopt linear model to classify new examples, because of which we could use distances between samples and separating hyperplane to represent confidence scores (Pang and Lee, 2004). The output  $d_i$  of SVM is a signed distance (negative = negative orientation) from hyperplane, we convert  $d_i$  to non-negative by equation (1).

$$P_{neg}(s) = \begin{cases} 1 & d_i > 1 \\ (1 + d_i)/2 & -1 \leq d_i \leq 1 \\ 0 & d_i < -1 \end{cases} \quad (1)$$

## 5 Experiments

This section details the experimental setup, including the corpora and lexicons we used, and the achieved results.

### 5.1 Experimental Setup

**Benchmark Datasets:** Our experiments are based on the Chinese Microblogs Sentiment Analysis Evaluation benchmark, China Computer Federation Conference on Natural Language Processing & Chinese Computer (NLP&CC2012). The evaluation is part of the NLP&CC2012, consisting 20 topics provided by Tencent Microblog, and there are 2207 subjective, 407 positive, and 1766 negative sentences.

**Sentiment Lexicon:** In our experiments, we integrate the following resources to construct a sentiment lexicon: (1) Sentiment lexicon provided by HowNet<sup>3</sup> which consists of 836 positive sentiment words and 1254 negative sentiment words; (2) NTU Sentiment Dictionary<sup>4</sup> from National Taiwan University. It contains 2,812 positive words and 8,276 negative words; (3) WI sentiment analysis

<sup>3</sup><http://www.keenage.com>

<sup>4</sup><http://nlg18.csie.ntu.edu.tw>

lexicon<sup>5</sup> constructed by Harbin Institute of Technology which consists of 1,428 sentiment words with sentiment scores.

**Feature selection** As described in section 3.2, microblogging services is different from traditional media such as blog and product reviews. Special features should be explored according to the characteristics of microblogs, the main features we used can be found in table.1.

Table 1: The main features for polarity classification of opinion sentences

NO.	Feature description	Example
1	sentiment words	good, bad
2	strength of sentiment words	strength of pleasure, anger, sorrow, fear
3	rhetorical structure	question
4	emoticons	^_^
5	preposition	it, he, she
6	slang	给力(geli) <sup>6</sup>
7	repeated punctuation	!!!, ???
8	condition operator relates to a sentiment statement	despite, however, negative operator

### 5.2 Experimental Results

#### Supervised Learning for Imbalanced Sentiment Classification of Chinese Microblogs

In this section, we perform SVM and MaxEnt as our basic polarity classifier for sentiment classification of Chinese microblogs. Downsampling and oversampling are two widely used resampling technique for imbalanced classification, thus for thorough comparison, we apply SVM and MaxEnt model based on full training, downsampling, and oversampling method, depicted as follows.

- 1) Full-training: using the entire labeled corpora for training.
- 2) Downsampling: drop some of the majority class samples at random to obtain a balanced data set.
- 3) Oversampling: randomly duplicate the minority class samples to keep balance between the majority class and minority class.

<sup>5</sup><http://wi.hit.edu.cn>

<sup>6</sup>“geili” is a Chinese word in English alphabet, which means something is cool, or cooperative.



Table 2: Performances of different methods for imbalanced sentiment classification

Approach		Evaluation metrics					
		Micro-average			Macro-average		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
SVM	full-training	0.8542	0.6364	0.7294	0.8581	0.6312	0.7246
	oversampling	0.8006	0.5965	0.6836	0.8007	0.5881	0.6754
	downsampling	0.8393	0.6253	0.7166	0.8432	0.6195	0.7116
Maximum Entropy	full-training	0.8899	0.6630	0.7598	0.8887	0.6527	0.7497
	oversampling	0.8869	0.6608	0.7573	0.8770	0.6461	0.7411
	downsampling	0.8452	0.6297	0.7217	0.8363	0.6231	0.7141
Self-training	full-training	0.8958	0.6674	0.7649	0.8938	0.6571	0.7544
	oversampling	0.8929	0.6652	0.7624	0.8877	0.6533	0.7497
	downsampling	0.8631	0.6430	0.7370	0.8542	0.6292	0.7217
CRFs (baseline)		0.8332	0.7172	0.7709	0.8296	0.7152	0.7682
<b>Reserved self-training</b>		0.9194	0.6829	0.7837	0.9134	0.6785	0.7786
<b>Reserved self-training with min. cuts</b>		0.9313	0.6918	0.7939	0.9254	0.6874	0.7888

Figure 2 shows the performance of supervised polarity classifiers for 20 topics based on different imbalanced classification methods. We employed MaxEnt for both self-training and reserved self-training in our subsequent experiments because MaxEnt performed better than SVM. Contrary to the results of Li et al. (2011) in which downsampling approach performed best, in our study, full-training performs at least not bad than downsampling and oversampling. We speculate that these 20 microblog topics involve multiple domains such as political, environmental, and health issues, it would lose some potentially useful information if downsampling method is applied, which induced a bad performance of downsampling. In addition, all the methods perform badly on topic 3, 6 and 11 in which positive sentences account for the major class as shown in Li et al. (2011). There are two possible reasons for these results: (1) training data set of the NLP&CC2012 is imbalanced, the number of negative sentences is 4 times that of positive one, which results in model's bias towards the majority class, namely negative sentences; (2) topic 3, 6 and 11 contain much more positive sentences than the others.

### Reserved Self-training for Imbalanced Sentiment Classification of Chinese Microblogs

In this subsection, we report the performance of reserved self-training on imbalanced sentiment classification of Chinese microblogs. We implemented a model that achieved the best run in

the NLP&CC2012 for comparison. It employed Conditional Random Fields(CRFs) to predict unlabeled data, and we treated this model as our evaluation baseline in our experiments.

The entire labeled training corpora is divided into three groups, a labeled training corpus, an unlabeled data set that is actually annotated in order to facilitate the experiments, and a reserved labeled sample set. We perform fivefold cross-validation and use the averaged results as our final estimation. In Figure 3, we can see that reserved self-training performed better than the other methods, especially on topic 3, 6, and 11 in which positive sentences accounted for the major class. A detail comparison of different methods can be found in Table 2. It is worth mentioning that incorporating context information by minimum cuts is able to enhance the performance of our results.

## 6 Conclusions and Future work

In this study, we focus on the problem of imbalanced sentiment classification of Chinese microblogs. Experiments show that reserved self-training could effectively make use of imbalanced labeled and unlabeled data to achieve better performance with less training data compared with full training, while downsampling and oversampling failed to make improvement. Additionally, combining the context information between different sentences based on minimum cuts is able to revise bad classification. Inspired by active learning,

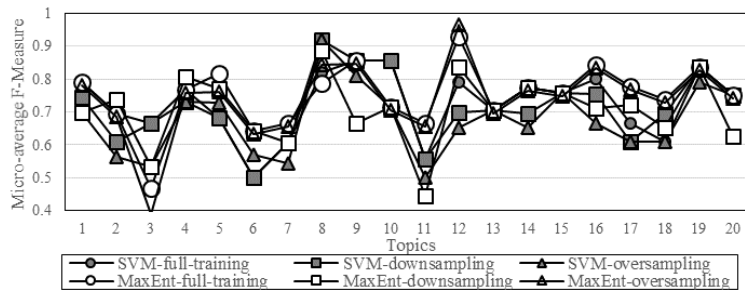


Figure 2: Performances of supervised polarity classifiers for different topics

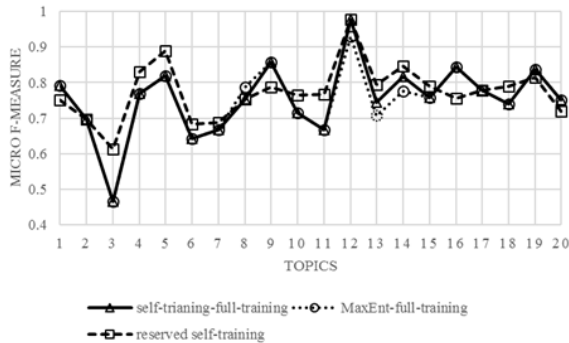


Figure 3: Comparison of different approaches with reserved self-training on imbalanced data

reserved self-training incorporate both the most confident unlabeled samples, together with their predicted labels, and the least confident labeled samples into training set. The classification error can be reduced because the least confident labeled samples would help the model better discriminate different classes. Thus, the selection strategy of reserved self-training can be applied to resolve other problems involving imbalanced binary classification, and not restricted to sentiment classification of microblogs. In the future, we will try to extend this method to address multi-label classification problems.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 90924015.

## References

Maria-Florina Balcan, Avrim Blum, and Ke Yang. 2004. Co-training and expansion: Towards bridging theory and practice. In *Advances in neural information processing systems*, pages 89–96.

Soumya Banerjee and Nitin Agarwal. 2012. Ana-

lyzing collective behavior from blogs using swarm intelligence. *Knowledge and information systems*, 33(3):523–547.

Ricardo Barandela, José Salvador Sánchez, Vicente Garcia, and Edgar Rangel. 2003. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851.

Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. *Urbana*, 51:61801.

Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. 2006. *Semi-supervised learning*, volume 2. MIT press Cambridge.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June.

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.

Chris Drummond, Robert C Holte, et al. 2003. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II*, volume 11. Citeseer.

Yen-Hsien Lee, Paul Jen-Hwa Hu, Tsang-Hsiang Cheng, and Ya-Fang Hsieh. 2012. A cost-sensitive technique for positive-example learning supporting content-based product recommendations in b-to-c e-commerce. *Decision Support Systems*, 53(1):245–256.

- Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. 2011. Semi-supervised learning for imbalanced sentiment classification. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1826–1831. AAAI Press.
- Charles X Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. 2004. Decision trees with minimal costs. In *Proceedings of the twenty-first international conference on Machine learning*, page 69. ACM.
- Bing Liu, Mingqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.
- Jonathan Ortigosa-Hernández, Juan Diego Rodríguez, Leandro Alzate, Manuel Lucania, Iñaki Inza, and Jose A Lozano. 2012. Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. *Neurocomputing*, 92:98–115.
- 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*, page 271. Association for Computational Linguistics.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.
- Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 25–32. Association for Computational Linguistics.
- Russell J Ryan. 2011. *Groundtruth budgeting: a novel approach to semi-supervised relation extraction in medical language*. Ph.D. thesis, Massachusetts Institute of Technology.
- Aarti Singh, Robert Nowak, and Xiaojin Zhu. 2008. Unlabeled data: Now it helps, now it doesn't. In *Advances in Neural Information Processing Systems*, pages 1513–1520.
- 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*, pages 417–424. Association for Computational Linguistics.
- Gary M Weiss. 2004. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19.
- Wei Wu and Mari Ostendorf. 2013. Graph-based query strategies for active learning.
- Show-Jane Yen and Yue-Shi Lee. 2009. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727.
- Bianca Zadrozny, John Langford, and Naoki Abe. 2003. Cost-sensitive learning by cost-proportionate example weighting. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 435–442. IEEE.

# Enhancing Lexicon-Based Review Classification by Merging and Revising Sentiment Dictionaries

**Heeryon Cho**

Yonsei Institute of Convergence  
Technology, Yonsei University  
Incheon, Republic of Korea

**Jong-Seok Lee    Songkuk Kim**

School of Integrated Technology  
Yonsei University  
Incheon, Republic of Korea

{heeryon, jong-seok.lee, songkuk}@yonsei.ac.kr

## Abstract

This paper presents a method of improving lexicon-based review classification by merging multiple sentiment dictionaries, and selectively removing and switching the contents of merged dictionaries. First, we compare the positive/negative book review classification performance of eight individual sentiment dictionaries. Then, we select the seven dictionaries with greater than 50% accuracy and combine their results using (1) averaging, (2) weighted-averaging, and (3) majority voting. We show that the combined dictionaries perform only slightly better than the best single dictionary (65.8%) achieving (1) 67.8%, (2) 67.7%, and (3) 68.3% respectively. To improve this, we combine seven dictionaries at a deeper level by merging the dictionary entry words and averaging the sentiment scores. Moreover, we leverage the skewed distribution of positive/negative threshold setting data to update the merged dictionary by selectively removing the dictionary entries that do not contribute to classification while switching the polarity of selected sentiment scores that hurts the classification performance. We show that the revised dictionary achieves 80.9% accuracy and outperforms both the individual dictionaries and the shallow dictionary combinations in the book review classification task.

## 1 Introduction

With the increase in opinion mining and sentiment analysis-related researches, various lexical resources that define sentiment scores/categories have been constructed and made available. Examples include SentiSense (de Albornoz *et al.*, 2012), SentiWordNet (Baccianella *et al.*, 2010), Micro-WNOp, and WordNet-Affect (Strapparava and Valitutti, 2004), which are based on a large English lexical database WordNet (Fellbaum,

1998), and AFINN (Nielsen, 2011), Opinion Lexicon (Hu and Liu, 2004), Subjectivity Lexicon (Riloff and Wiebe, 2003) and General Inquirer (Stone and Hunt, 1963), which are manually or semi-automatically constructed. These resources differ in their formats and sizes, but all can be utilized in the lexicon-based opinion mining and sentiment analysis.

The increase in the number of sentiment resources naturally gives rise to two questions: (1) How are the performances of these resources different? (2) Can we construct a better sentiment resource by combining and/or revising multiple resources? We answer these questions by comparing the book review classification performance of single and combined sentiment resources, and present a simple ‘merge, remove, and switch’ approach that revises the entries of the sentiment resource to improve its classification performance.

In the next section, we describe the experimental setup for evaluating the classification performance of sentiment resources. We then compare the positive/negative classification performance of eight widely known individual sentiment resources in Section 3. Since individual sentiment resources are originally constructed in different formats, we standardize their formats. These standardized resources will be called *sentiment dictionaries* or simply *dictionaries* throughout this paper. In Section 4 we compare the classification performances of combined dictionaries which integrate multiple individual dictionaries’ results using averaging, weighted-averaging, and majority voting. Then, we introduce a method of revising sentiment dictionaries at a deeper level by merging, removing, and switching the dictionary contents. Implications for utilizing multiple dictionaries are discussed. Related works are introduced in Section 5, and conclusion is given in Section 6.

## 2 Experimental Setup

90,000 Amazon book reviews were collected to construct a positive/negative review dataset for sentiment dictionary evaluation.

### 2.1 Dataset

5-star and 4-star book reviews were merged and labeled as positive reviews, and 1-star and 2-star reviews were merged and labeled as negative reviews. 3-star reviews were excluded. 10,000 reviews (positive reviews: 9,007 / negative reviews: 993) were randomly selected as positive/negative threshold setting data (see 2.3). The remaining 80,000 reviews (positive reviews: 71,993 / negative reviews: 8,007) were set aside as test data.

### 2.2 Review Sentiment Score Calculation

Eight sentiment resources (see Table 1) were standardized to generate eight sentiment diction-

aries ( $D_j, j=1, \dots, d$ ). (The standardization of sentiment resources is discussed in Section 3.1.)

Each book review was tokenized, lemmatized, and part-of-speech tagged using the Stanford CoreNLP suite (Toutanova *et al.*, 2003). Once the list of words in the review was obtained, the sentiment score ( $D_j(w_i)$ ) of each word was looked up in the sentiment dictionary ( $D_j$ ). The scores of all the review words listed in the dictionary ( $w_i, i=1, \dots, n$ ) were averaged to yield the *Review Sentiment Score (RSS)*.

$$RSS(D_j) = \frac{1}{n} \sum_{i=1}^n D_j(w_i)$$

Because the Stanford Part-of-Speech Tagger outputs detailed parts of speech whereas the standardized sentiment dictionaries either do not define or define only four parts of speech (e.g., noun, adjective, verb, and adverb), the Tagger's parts of speech were mapped to four parts of speech as shown in Table 3.

Resource	Entry Size	Sentiment Category & Score Range	Note
AFINN <sup>1</sup>	2,477 words	No categories. Each word has integer score ranging between -5 (very negative) and 5 (very positive).	Based on Affective Norms for English Words (ANEW).
General Inquirer <sup>2</sup>	11,788 words	Positiv/Negativ/Pstv/Ngtv/Pleasur/Pain/EMOT/etc. categories. No numerical scores.	Based on Harvard IV-4 and Lasswell dictionaries, etc.
Micro-WNOP <sup>3</sup>	1,105 synsets/ 1,960 words	Positive/negative/objective categories each with 0~1 score.	Based on WordNet 2.0.
Opinion Lexicon <sup>4</sup>	6,786 words	Positive/negative categories. No numerical scores.	Misspelled words are deliberately included.
SentiSense <sup>5</sup>	2,190 synsets/ 4,404 words	Joy/sadness/love/hate/despair/hope/etc. 14 emotion categories. No numerical scores.	Based on WordNet 2.1.
SentiWordNet <sup>6</sup>	117,659 synsets/ 155,287 words	Positive/negative/objective categories each with 0~1 score.	SentiWordNet ver. 3.0. Based on WordNet 3.0.
Subjectivity Lexicon <sup>7</sup>	8,221 words	Positive/negative/both/neutral categories. No numerical scores.	Subjectivity (weak/strong) is also defined.
WordNet-Affect <sup>8</sup>	2,872 synsets/ 4,552 words	Synsets are first categorized into emotion/mood/trait/behavior/etc., and these categories are further categorized into positive/negative/ambiguous/neutral. No numerical scores.	Based on WordNet 1.6.

Table 1. The contents of eight sentiment resources.

<sup>1</sup> [http://www2.imm.dtu.dk/pubdb/views/publication\\_details.php?id=6010](http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010)

<sup>2</sup> [http://www.wjh.harvard.edu/~inquirer/spreadsheet\\_guide.htm](http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm)

<sup>3</sup> <http://www-3.unipv.it/wnop/>

<sup>4</sup> <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon>

<sup>5</sup> <http://nlp.uned.es/~jcalbornoz/resources.html>

<sup>6</sup> <http://sentiwordnet.isti.cnr.it/>

<sup>7</sup> [http://mpqa.cs.pitt.edu/lexicons/subj\\_lexicon/](http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/)

<sup>8</sup> <http://wndomains.fbk.eu/wnaffect.html>

	AFN <sup>1</sup>	GI <sup>2</sup>	MWO <sup>3</sup>	OL <sup>4</sup>	SS <sup>5</sup>	SWN <sup>6</sup>	SL <sup>7</sup>	WNA <sup>8</sup>
AFN <sup>1</sup>	<u>2,477</u> <b>2,454</b> <i>1,723</i>							
GI <sup>2</sup>	<b>917</b> <i>913</i>	<u>11,788</u> <b>3,906</b> <i>3,853</i>						
MWO <sup>3</sup>	<b>196</b> <i>190</i>	<b>551</b> <i>551</i>	<u>1,960</u> <b>1,515</b> <i>1,334</i>					
OL <sup>4</sup>	<b>1,315</b> <i>1,148</i>	<b>2,504</b> <i>2,485</i>	<b>470</b> <i>465</i>	<u>6,786</u> <b>6,560</b> <i>5,393</i>				
SS <sup>5</sup>	<b>771</b> <i>742</i>	<b>1,238</b> <i>1,237</i>	<b>397</b> <i>375</i>	<b>1,533</b> <i>1,476</i>	<u>4,404</u> <b>3,729</b> <i>3,225</i>			
SWN <sup>6</sup>	<b>1,781</b> <i>1,615</i>	<b>3,870</b> <i>3,836</i>	<b>1,504</b> <i>1,330</i>	<b>5,386</b> <i>5,080</i>	<b>3,715</b> <i>3,217</i>	<u>155,287</u> <b>77,761</b> <i>33,923</i>		
SL <sup>7</sup>	<b>1,246</b> <i>1,182</i>	<b>3,047</b> <i>3,021</i>	<b>586</b> <i>582</i>	<b>5,296</b> <i>4,771</i>	<b>1,738</b> <i>1,685</i>	<b>6,130</b> <i>5,860</i>	<u>8,221</u> <b>6,731</b> <i>6,059</i>	
WNA <sup>8</sup>	<b>312</b> <i>292</i>	<b>391</b> <i>391</i>	<b>122</b> <i>118</i>	<b>835</b> <i>550</i>	<b>938</b> <i>786</i>	<b>1,024</b> <i>857</i>	<b>639</b> <i>609</i>	<u>4,552</u> <b>1,035</b> <i>864</i>

Table 2. Number of shared single word entries disregarding the parts of speech between two dictionaries (**bold numbers**). Numbers in *italics* are the actual dictionary entries that match the book review words. The underlined numbers in the diagonal cells are the actual entry word size of each dictionary.

<sup>1</sup>AFINN, <sup>2</sup>General Inquirer, <sup>3</sup>Micro-WNOp, <sup>4</sup>Opinion Lexicon, <sup>5</sup>SentiSense, <sup>6</sup>SentiWordNet, <sup>7</sup>Subjectivity Lexicon, <sup>8</sup>WordNet-Affect.

### 2.3 Threshold Setting & Judgment

Each dictionary’s threshold for judging the positivity and negativity (i.e., the review label) of the book reviews was set using the threshold setting data; the threshold with the greatest accuracy was selected. A review was judged as positive if the *RSS* was greater than or equal to the threshold, and as negative, otherwise.

$$ReviewLabel = \begin{cases} \text{positive} & \text{if } RSS \geq \text{threshold} \\ \text{negative} & \text{otherwise} \end{cases}$$

### 2.4 Performance Measure

Since the book review dataset was an imbalanced dataset containing more positive reviews than negative reviews (i.e., 9:1 ratio), balanced accuracy ( $Acc_{BAL}$ ) was used to measure the overall performance.

$$Acc_{BAL} = 0.5 \times Recall_{POS} + 0.5 \times Recall_{NEG}$$

$$Recall_{POS} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$Recall_{NEG} = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$$

$Recall_{POS}$  and  $Recall_{NEG}$  each measure the positive and negative review accuracy.

Stanford POS Tagger	Senti.Dict.
NN, NNS, NNP, NNPS	Noun
JJ, JJR, JJS	Adjective
VB, VBD, VBG, VBN, VBP, VBZ	Verb
RB, RBR, RBS	Adverb

Table 3. Part-of-speech mapping from Stanford POS Tagger to the sentiment dictionary.

### 3 Individual Dictionary Comparison

The bold numbers in Table 2 indicate the number of shared *single words* between two sentiment dictionaries; note that the part-of-speech was disregarded when extracting the shared words. The underlined numbers in the diagonal cells are the actual dictionary entry word sizes. The italicized numbers are the dictionary entry words that actually match the book review words.

The eight sentiment dictionaries in Table 2 all include the following thirty-one words: “approval”, “cheer”, “cheerful”, “contempt”, “cynical”, “disdain”, “earnest”, “excitement”, “fantastic”, “glee”, “gloomy”, “good”, “guilt”, “horrible”, “marvel”, “offend”, “proud”, “reject”, “scorn”, “sick”, “sincerity”, “sore”, “sorrow”, “sorry”, “triumph”, “trouble”, “ugly”, “upset”, “vile”, “warm”, and “worry”.

Dictionary	Recall <sub>POS</sub>	Recall <sub>NEG</sub>	Acc <sub>BAL</sub>	Threshold
AFINN	59.6%	67.9%	63.8%	0.140
General Inquirer	62.2%	69.3%	<b>65.8%</b>	0.175
Micro-WNOp	40.7%	70.0%	55.4%	0.120
Opinion Lexicon	<b>67.0%</b>	63.4%	65.2%	0.025
SentiSense	61.2%	64.3%	62.8%	0.225
SentiWordNet	<b>67.0%</b>	64.3%	65.7%	0.005
Subjectivity Lexicon	58.7%	<b>70.6%</b>	64.7%	0.170
WordNet-Affect	59.8%	38.2%	49.0%	0.005

Table 4. Positive ( $Recall_{POS}$ ), negative ( $Recall_{NEG}$ ), and balanced accuracy ( $Acc_{BAL}$ ) of eight sentiment dictionaries on 80,000 book reviews.

### 3.1 Standardization

Because some sentiment resources define sentiment *categories* instead of sentiment *scores*, they were converted to sentiment scores: For example, *positive*, *negative*, and *neutral/ambiguous* categories were each converted to 1.0, -1.0, and 0.0.

In some cases, emotion categories such as *joy*, *sadness*, *love*, etc. were first mapped to *positive*, *negative*, or *ambiguous* categories and then converted to sentiment scores. The standardization process for each dictionary is explained below.

**AFINN:** AFINN contains sentiment scores ranging between  $-5 \leq score_{AFINN} \leq 5$ . These scores were normalized from  $[-5..5]$  to  $[-1..1]$ .

Normalizing  $[A..B]$  to  $[C..D]$  employed the following equation:

$$X' = \frac{D - C}{B - A} \cdot X + \frac{C \times B - A \times D}{B - A}$$

The below equation was used for AFINN:

$$X' = 0.2X$$

**General Inquirer (GI):** Each entry word in the GI contains one or more GI categories, and we selected the following sentiment-related categories and calculated the sentiment scores by averaging the assigned category values: *Positiv*, *Pstiv*, *PosAff*, *Pleasur*, *Virtue*, *Complet*, and *Yes* categories were each assigned a 1.0 score while *Negativ*, *Ngtv*, *NegAff*, *Pain*, *Vice*, *Fail*, *No*, and *Negate* categories were assigned a -1.0 score.

**Micro-WNOp:** For each entry word, the positive/negative paired sentiment scores were given by multiple human judges. These paired scores were added and averaged to obtain a single sentiment score. Note that for all WordNet-based sentiment resources, the different senses of a word (e.g., *happy#1*, *happy#2*, etc.) were aggregated and their sentiment scores were averaged.

**Opinion Lexicon:** Words in the positive word list were given a 1.0 score while words in the negative word list were given a -1.0 score. Three ambiguous words that were included in both the positive and negative lists were given a 0.0 score.

**SentiSense:** Emotional categories assigned to the synsets were converted to sentiment scores: *Joy*, *love*, *hope*, *calmness*, and *like* categories were given a 1.0 score; *fear*, *anger*, *disgust*, *surprise*, and *anticipation* categories were given a -1.0 score; and *ambiguous*, *surprise*, and *anticipation* categories were given a 0.0 score.

**Subjectivity Lexicon:** *Positive*, *negative* and *neutral* categories were converted to 1.0, -1.0, and 0.0 sentiment scores respectively. Entry words with ‘anypos’ (i.e., any parts-of-speech) were unfolded to have four parts-of-speech.

**WordNet-Affect:** Synsets having affective hierarchical categories such as *positive-emotion*, *negative-emotion*, *ambiguous-emotion*, and *neutral-emotion* were converted to 1.0, -1.0, 0.0, and 0.0 sentiment scores respectively.

Note that only the single word dictionary entries were actually looked up in the book review classification experiments; phrases or compound words (e.g., those including blank spaces, hyphens or underscores) were not matched.

### 3.2 Evaluation

The RSSs were calculated using the eight standardized sentiment dictionaries for each review, and the threshold for judging the review label was set differently for each dictionary using the 10,000 book review threshold setting data.

Table 4 compares the classification performance of the eight sentiment dictionaries on test data (80,000 book reviews).  $Recall_{POS}$ ,  $Recall_{NEG}$ , and  $Acc_{BAL}$  each indicate the classification accuracy of positive, negative, and overall reviews. Here, General Inquirer showed the best overall performance ( $Acc_{BAL}=65.8\%$ ); Opinion Lexicon and SentiWordNet performed well on positive reviews ( $Recall_{POS}=67.0\%$ ) whereas Subjectivity Lexicon performed well on negative reviews ( $Recall_{NEG}=70.6\%$ ).

Despite the significant difference between the General Inquirer and SentiWordNet’s book review-related dictionary entry word sizes (Table

2: 3,853 vs. 33,923), the two exhibited comparable classification accuracies. The same can be said for the rest of the dictionaries excluding the lowest performing two dictionaries, MicroWNOp and WordNet-Affect.

#### 4 Combining Multiple Dictionaries

We now investigate the performance of combining multiple dictionaries through averaging, weighted-averaging, and majority voting.

##### 4.1 McNemar’s Test

We applied McNemar’s test (McNemar, 1947) on the classification results of the individual sentiment dictionaries to investigate whether any two dictionaries’ hits and misses were significantly different. The worst performing WordNet-Affect was excluded from the test.

Twenty-one dictionary pairs were generated from seven sentiment dictionaries. All dictionary pairs except the Opinion Lexicon vs. SentiWordNet ( $p=0.5552$ ) exhibited significant differences in the proportion of hits and misses at 5% significance level<sup>1</sup>.

##### 4.2 Averaging, Weighted-Averaging, & Majority Voting

**Averaging:** The seven dictionaries’  $RSS$ s were averaged for each book review to calculate the combined *Averaged Review Sentiment Score* ( $RSS_{AVG}$ ). We excluded the worst performing WordNet-Affect with lower than 50% accuracy since classifiers involved should provide a lower error rate than a random classifier (Enrriquez *et al.*, 2013).

$D_j$  indicates the individual dictionary,  $j$  denotes the index of the sentiment dictionary, and  $m$  indicates the number of sentiment dictionaries to be combined; in our case  $m$  equals seven.

$$RSS_{AVG} = \frac{1}{m} \sum_{j=1}^m RSS(D_j)$$

$$ReviewLabel = \begin{cases} \text{positive} & \text{if } RSS_{AVG} \geq thres \\ \text{negative} & \text{otherwise} \end{cases}$$

A review was judged as positive if the  $RSS_{AVG}$  was greater than or equal to the threshold, and as negative, otherwise. The threshold was determined using the threshold setting data.

	Recall <sub>POS</sub>	Recall <sub>NEG</sub>	Acc <sub>BAL</sub>	Thres
AVG	66.7%	68.9%	67.8%	0.115
w-AVG	<b>67.3%</b>	68.0%	67.7%	0.045
Vote	64.1%	<b>72.5%</b>	<b>68.3%</b>	N/A

Table 5. Classification accuracy of the combined dictionaries using averaging (AVG), weighted-averaging (w-AVG), and majority voting (Vote).

**Weighted-Averaging:** The seven dictionaries’  $RSS$ s were weighted and averaged to yield a combined *Weighted-Averaged Review Sentiment Score* ( $RSS_{w-AVG}$ ).

$$RSS_{w-AVG}(weight_j) = \frac{1}{m} \sum_{j=1}^m weight_j \cdot RSS(D_j)$$

$$\sum_j weight_j = 1, (0 \leq weight_j \leq 1)$$

$$ReviewLabel = \begin{cases} \text{positive} & \text{if } RSS_{w-AVG} \geq thres \\ \text{negative} & \text{otherwise} \end{cases}$$

Grid search was performed to set the weights of the seven dictionaries during the threshold setting stage. In the experiment, AFINN was given the greatest weight of 0.4, while the remaining six dictionaries were each given 0.1 weights.

**Majority Voting:** The classification result (label) of each sentiment dictionary was used as votes in the majority voting. In the case of voting, the threshold was not set.

$$ReviewLabel = \begin{cases} \text{positive} & \text{if } Vote_{pos} > Vote_{neg} \\ \text{negative} & \text{otherwise} \end{cases}$$

Table 5 compares the classification accuracy of the three combined dictionaries on test data. AVG, w-AVG, and Vote each indicate averaging, weighted-averaging, and majority voting. The majority voting showed the best performance on the negative (72.5%) and overall (68.3%) review classification while the weighted-averaging showed the best performance on the positive (67.3%) review classification. However, the performance increase of the combined method was marginal compared to the best performing single dictionary (General Inquirer’s 65.8% vs. majority voting’s 68.3%).

##### 4.3 Merging, Removing, & Switching

Combining multiple dictionaries at the surface level did not bring much improvement. We decided to merge the dictionaries at a deeper level and revise the dictionary entry’s sentiment scores to improve the classification performance.

<sup>1</sup> AFINN vs. SentiSense ( $p=5.221e-09$ ), AFINN vs. Subjectivity Lexicon ( $p=0.0002395$ ), General Inquirer vs. SentiSense ( $p=2.886e-12$ ), and the remaining seventeen dictionary pairs ( $p<2.2e-16$ ).



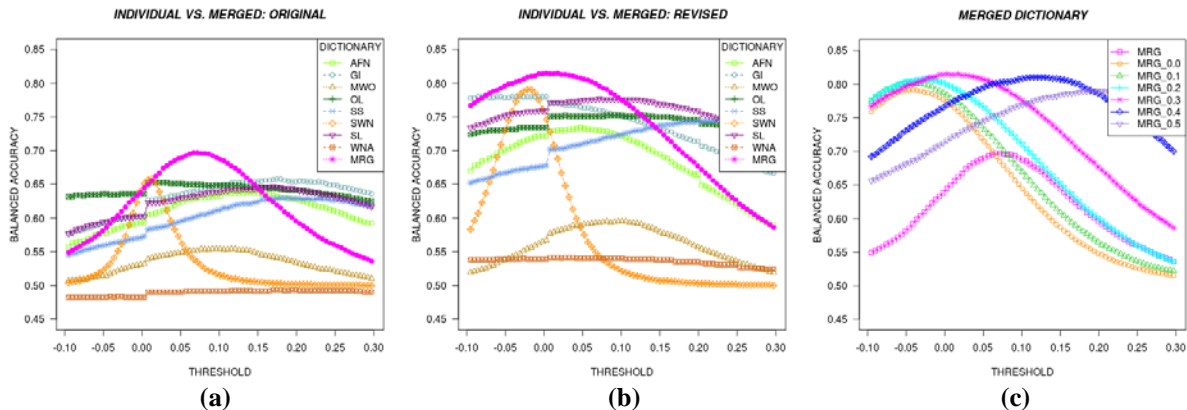


Figure 1. Classification performance of (a) eight original individual dictionaries (*AFN~WNA*) and one merged dictionary (*MRG*) and (b) revised dictionaries across different thresholds. (c) Classification performance of original merged dictionary (*MRG*) and revised merged dictionaries with different values of ‘remove & switch’ (*MRG\_0.0~MRG\_0.5*).

We could have merged all eight dictionaries, but instead merged the seven dictionaries excluding the SentiWordNet; we guessed that adding the largest SentiWordNet would simply result in an expanded version of the SentiWordNet and similar performance to the SentiWordNet. When merging the dictionary entries, the sentiment scores of the overlapping entry words, disregarding the parts-of-speech, were averaged. As a result, a merged sentiment dictionary containing 12,114 word entries was created. Threshold was also set for the merged dictionary, and the 80,000 book reviews were classified.

Table 6 compares the classification performance of the individual (*AFN~WNA*) and merged dictionaries (*MRG*). The first column lists the dictionaries (see Table 2 bottom for the full names of the dictionaries.), the second column displays the performance of the original dictionaries, and the third column shows the performance of the revised dictionaries. We confirmed that the merged dictionary (*MRG*) showed better performance (69.5%) than both the individual dictionaries and the best performing combined dictionary using majority voting (68.3%). Figure 1 (a) compares the performance of the nine dictionaries across different thresholds. We see that the merged dictionary (pink curve) outperforms the rest between the 0.05~0.10 threshold ranges.

Still, the performance of the merged dictionary did not improve dramatically. Therefore, we contrived a way to update the merged dictionary’s entries to enhance performance. To do this, we leveraged the skewed distribution of positive/negative reviews. The general idea is to selectively (1) remove dictionary entries and (2) switch the polarity of sentiment scores.

Senti. Dict.	Original $Acc_{BAL}$ (thres)	Revised $Acc_{BAL}$ (thres)
<b>AFN</b>	63.8% (0.140)	73.2% (-0.030)
<b>GI</b>	65.8% (0.175)	78.0% (-0.085)
<b>MWO</b>	55.4% (0.125)	58.9% (0.045)
<b>OL</b>	65.2% (0.025)	75.1% (0.015)
<b>SS</b>	62.8% (0.225)	72.0% (0.085)
<b>SWN</b>	65.7% (0.005)	78.8% (-0.030)
<b>SL</b>	64.7% (0.170)	77.2% (0.005)
<b>WNA</b>	49.0% (0.005)	54.0% (0.075)
<b>MRG</b>	<b>69.5% (0.060)</b>	<b>80.9% (-0.025)</b>

Table 6. Classification performance of original and revised dictionaries and their thresholds.

To implement the first idea, we removed those dictionary entry words with positive/negative book review word occurrence ratios that are similar to that of positive/negative book review ratio itself. The selection of the word was determined using the threshold setting data. For example, if the word “interested” appeared in the positive and negative reviews 900 and 100 times respectively, and the positive/negative review ratio of the threshold setting data is 9:1, we removed the “interested” entry from the dictionary. Such entry words were considered as not contributing to the actual classification.

To implement the second idea, we switched the sign of the selected dictionary entry words’ sentiment scores whose positive/negative word occurrence ratio and the positive/negative review ratio’s difference yielded a value with the sign opposite of its sentiment scores. For example, if the word “horror” appeared 900 and 300 times in the positive/negative book reviews resulting in 9:3 word occurrence ratios and the review ratio itself is 9:1, we calculated the difference between

the word and review ratio as  $9/3 - 9/1$ , which resulted in  $9/2$ , a positive number. However, the sentiment dictionary originally lists the sentiment score of “horror” as negative, e.g.,  $-0.858$ ; hence the sign (polarity) of the entry word “horror” was switched to positive, e.g.,  $0.858$ .

Table 6 shows the classification performance of the revised dictionaries. We see that the performance increased for all original dictionaries after they were revised using the ‘remove & switch’ procedure. The merged and revised dictionary showed the best performance (80.9%). Figure 1 (b) shows the performance of the revised dictionaries across different thresholds. We see that our method works better with larger dictionaries than smaller dictionaries such as *MWO* and *WNA*. This may be natural since our method includes the ‘remove’ procedure.

How much dictionary contents to ‘remove & switch’ were determined using the threshold setting data by experimenting with different proportion values. Figure 1 (c) compares the merged dictionary in its original version (*MRG*) and the revised versions using different values for revising (*MRG\_0.0~MRG\_0.5*). In our experiment, the best performing merged and revised dictionary’s ‘remove & switch’ value was determined as 0.3 (*MRG\_0.3*).

Our approach employs the most basic sentiment score aggregation to perform classification; no negation handling or structural analysis of the sentences is conducted. Our focus is on revising the sentiment dictionary by utilizing multiple dictionaries. At the outset, we surmised that combining and revising multiple dictionaries will have the following effects: (1) the word coverage will broaden and different dictionaries will complement each other. (2) The sentiment scores will be updated to incorporate diverse measurements leading to less odd scores.

However, broader coverage did not necessarily guarantee better performance since irrelevant words often matched to generate noise. By incorporating the ‘remove’ procedure, we aimed to remove noise. Examples of removed words in the book reviews dataset included “book”, “interested”, and “mystery”. With regard to the assumption (2) above, we found that contextual adjustment of sentiment scores was necessary for the given domain. Consequently we proposed the ‘switch’ procedure which switched the polarity of selected dictionary entries. Examples of the switched words included “conspiracy”, “horror”, and “tragic” which were changed to have positive polarity.

## 5 Related Work

We were motivated by Taboada *et al.*’s (2011) work on lexicon-based sentiment analysis which couples hand-crafted sentiment dictionary with detailed sentence analysis. Although their sentiment calculation (SO-CAL) is more advanced than ours (it incorporates, for example, negation and intensification), we were able to confirm through the ‘remove’ procedure that “less is more”, i.e., less confounding dictionary entries will lead to more (greater) performance, with regard to the treatment of dictionary (Taboada *et al.*, 2011; p.297). Our contribution is that we provided a simple data-based method to achieve “less is more” by leveraging the skewed distribution of the threshold setting positive/negative review data. This will be useful when ample threshold setting data is available, but dictionary expert is absent or costly.

Fahrni and Klenner (2008) proposed domain-specific adaptation of sentiment-bearing adjectives. Adjectives (e.g., good, bad, etc.) possess prior polarity, but depending on the context this polarity may change; for instance, warm mittens may be desirable, but warm beer may not be. To tackle the problem of contextual polarity, Fahrni and Klenner implemented a two-stage process that first identifies domain-specific targets using Wikipedia, and then determines the target-specific polarity of adjectives using a corpus. We performed a crude polarity adaptation by selectively switching the polarity of the dictionary entry’s sentiment score based on the positive/negative distribution of the threshold setting data. Our approach, albeit crude, takes into account all dictionary entries, not restricted to adjectives, as candidates for polarity adaptation.

Neviarouskaya *et al.* (2011) described methods for automatically building and scoring new words based on sentiment-scored lemmas and types of affixes to create a sophisticated sentiment dictionary. Although we did not build sentiment dictionary from scratch, we experimented with shallow combinations and entry word merging of multiple dictionaries to show that shallow combination is insufficient, and that deeper-level merging and revising could be used as a viable method for enhancing the dictionary; in the process we generated revised dictionaries.

Various sentiment resources are built to perform different sentiment analysis tasks, so uniformly standardizing each resource may be unjust for some resources; moreover, we restrict our method’s effectiveness within the sentiment

analysis of product reviews which is considered to be an easier problem compared to shorter texts such as microblogs (Cambria *et al.*, 2013); we acknowledge these as our limitations.

## 6 Conclusion

We presented a method of merging multiple dictionaries, and removing and switching the merged dictionary's contents to achieve greater accuracy in the lexicon-based book review classification. In the future, we plan to investigate whether our approach is robust across different domains, how much threshold setting data is needed to achieve improvement in the revised dictionary, and what effects different positive/negative data distribution has on our method. We also plan to cover other sentiment resources such as SenticNet (Cambria *et al.*, 2010) in the future.

## Acknowledgments

This research was supported by the Korean Ministry of Science, ICT and Future Planning (MSIP) under the "IT Consilience Creative Program" supervised by the National IT Industry Promotion Agency (NIPA) of Republic of Korea. (NIPA-2013-H0203-13-1002)

## References

- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation*.
- Erik Cambria, Robert Speer, Catherine Havasi, and Amir Hussain. 2010. SenticNet: A publicly available semantic resource for opinion mining. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Fall Symposium*.
- Erik Cambria, Björn Schuller, Yungqing Xia, and Catherine Havasi. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15-21.
- Jorge Carrillo de Albornoz, Laura Plaza, and Pablo Gervás. 2012. SentiSense: An easily scalable concept-based affective lexicon for Sentiment Analysis. In *Proceedings of the International Conference on Language Resources and Evaluation*.
- Fernando Enríquez, Fermín L. Cruz, F. Javier Ortega, Carlos, G. Vallejo, and José A. Troyano. 2013. A comparative study of classifier combination applied to NLP tasks. *Information Fusion*, 14(3):255-267.
- Angela Fahrni and Manfred Klenner. 2008. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proceedings of Symposium on Affective Language in Human and Machine, AISB 2008 Convention*.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153-157.
- Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2011. SentiFul: A lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 2(1):22-36.
- Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC Workshop on Making Sense of Microposts*.
- Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Philip J. Stone and Earl B. Hunt. 1963. A computer approach to content analysis: Studies using the General Inquirer system. In *Proceedings of the American Federation of Information Processing Societies, Spring Joint Computer Conference*.
- Carlo Strapparava and Alessandro Valitutti. 2004. WordNet-Affect: An affective extension of WordNet. In *Proceedings of the International Conference on Language Resources and Evaluation*.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberley Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2): 267-307.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

# Exploring the Effects of Word Roots for Arabic Sentiment Analysis

**Shereen M. Oraby**

College of Engineering  
and Technology  
AAST, Alexandria, Egypt  
shereen.oraby@aast.edu

**Yasser El-Sonbaty**

College of Computing  
and Information Technology  
AAST, Alexandria, Egypt  
yasser@aast.edu

**Mohamad Abou El-Nasr**

College of Engineering  
and Technology  
AAST, Alexandria, Egypt  
mnasr@aast.edu

## Abstract

The inherent morphological complexity of languages such as Arabic entails the exploration of language traits that could be valuable to the task of detecting and classifying sentiment within text. This paper investigates the relevance of using the roots of words as input features into a sentiment analysis system under two distinct domains, in order to tailor the task more suitably to morphologically-rich languages such as Arabic. Different word-rooting solutions are employed in conjunction with a basic sentiment classifier, in order to demonstrate the potential of mapping Arabic words to basic roots for a language-specific development to the sentiment analysis task, showing a noteworthy improvement to baseline performance.

## 1 Introduction

An increasing need for quick and effective analysis of huge masses of text has sparked a revolution in the requirements of natural language processing systems, demanding an ability to handle varied types and formats of textual data for a wide range of language analysis tasks, both on the syntactic and semantic levels. The task of sentiment analysis in particular presents a unique form of text analytics due to the flourish of new opinionated web data in social media and otherwise, dealing with the detection and of opinions within a text, and then further with distinguishing their polarity.

Two main tasks are of great importance with respect to the classification of opinions in text, regardless of the language under inspection: the tasks of *subjectivity detection* in a set of statements to differentiate between purely *objective* reporting of information in the form of facts, as opposed to a *subjective* account of the information; and the

task of *sentiment analysis*, which entails classifying the resultant subjective statements into a set of classes, *positive*, *negative*, and *neutral*, depending on the polarity of the opinion expressed. With respect to the level of analysis performed, individual tasks may be more relevant than others: while subjectivity analysis is relevant at the sentence level to sort out opinionated statements, sentiment analysis can be appropriate at both the sentence and the document level, if the excerpts are already defined to be subjective, and the task is to distinguish the *polarity* of the opinion being expressed (Liu, 2012).

While much research has been attributed to the task of sentiment analysis in English, fewer attempts tackle the task in other more morphologically complex languages such as Arabic, and increasing amounts of information available in these languages makes the task of Arabic sentiment analysis a very relevant one, albeit a challenge for classification systems. Such language processing tasks are made more difficult in Arabic due to the lack of resources and tools available as well, despite a growing user and content base in the language.

This paper explores the implications of reducing words to their roots in order to find common, basic, sentiment-bearing components that will relate many words to a single source, and thus help to classify a larger number of words to aid in the Arabic sentiment analysis task. This is presented through comparisons within two distinct datasets where opinions are classified based on their sentiment using roots derived from three different rooting libraries. Section 2 discusses related background information on Arabic language morphology, and some intuition behind the use of rooting as an aid to such a classification task. Section 3 details related work in the fields of subjectivity and sentiment analysis. The root-based methodology proposed is presented in Section 4, followed by

results, evaluation, and analysis of the performed experiments in Section 5. Section 6 presents conclusions and possible directions for future work.

## 2 Background Information

The challenges behind the natural language processing of languages such as Arabic stem from rich morphologies, or internal word structures (Habash, 2008), and the intricate construction of words from roots and patterns specific to the language’s grammar. While Modern Standard Arabic (MSA) is the standard form of communication for written and broadcasted Arabic (Ryding, 2006), spoken Arabic exists in the form of many different dialects, all of which diverge significantly from written MSA (Habash, 2008). This makes standardizing language processing tasks in Arabic even more complicated, in addition to the problem of diacritization and text normalization for data retrieved from unregulated sources, which is often the case for the mining of data appropriate for tasks such as sentiment analysis. The following section gives a basic outline of some of the details of Arabic grammar and morphology relevant to the opinion classification task at hand as background for the proposed algorithm.

### 2.1 Roots in Morphologically-Rich Languages

In derivational languages such as Arabic, words are derived from sets of “roots”, which are commonly two, three, or four letter words that describe a basic idea (StudyQuran, 2004). Full words in Arabic are then derived from these roots by adding vowels (and/or other consonants) around the basic root, called *affixes*, which change the word pronunciation and form word derivations (Albraheem and Al-Khalifa, 2012; Ryding, 2006). These affixes can be attached to a base, stem, or root, as either *prefixes* (inserted before the word), *infixes* (inserted within the word), or *suffixes* (inserted after the word).

As an example, the Arabic letters س ل م (*siin-*

*laam-miim*) serve as the root for several words, including *salām*, *īsalām*, and *muslim* (StudyQuran, 2004), as shown in Table 1. By sharing the same root word, these three derivations also share a common base meaning. This pattern is a result of the word formation scheme in Arabic, where a root such as ك ت ب (*kaaf-ta’-ba’*) means having to do with “writing”, and where most other Arabic words “having to do with writing” are derived from additions and modifications to this basic three-letter root, such as كتاب (*kitāb*, meaning *book*), كاتب (*kātib*, meaning *writer*), مكتب (*maktab*, meaning *desk*), and مكتبة (*maktabatu*, meaning *library*) (Ryding, 2006).

The idea of words carrying meaning from their basic root derivatives is different to the system of word derivation in concept-based languages, where only some subset of words, but not most, can be likened to the root system. Such patterns do exist in languages such as English, but are not a general rule for word derivation. For example, the English “consonant sequence” *s-ng*, which can be used to compose various derived words, including *s-i-ng*, *s-a-ng*, *s-u-ng*, and *s-o-ng*, each of which shares a common base meaning having to do with “vocal music”. Attaching various prefixes and suffixes to these derivations also results in a wider array of words, including *sing-ing*, *sing-er*, and *unsung* (Ryding, 2006).

This concept maps the English consonant sequence to the Arabic *root*, and the English derivations resultant from the addition of vowels and affixes to the concept of an Arabic *pattern*. The consistence of this word-derivation scheme across most of a language gives root-based languages such as Arabic a well-defined clarity for word formation, which could be used to classify words based on common meaning or sense.

### 2.2 Intuition Behind Root-Based Matching

Due to this word formulation and root-based derivation scheme that is prevalent Arabic, many words bearing similar meanings come from the

Arabic Word	Transliteration	English Gloss
سلام	<i>salām</i>	peace
إسلام	<i>īslām</i>	submission, compliance, conformance, surrender
مسلم	<i>muslim</i>	one who submits, complies, conforms, surrenders

Table 1: Various Arabic Word Derivations for the Root Word *Siin-Laam-Miim* (StudyQuran, 2004).

same root, which in itself holds the “idea” that the derivations express.

It is this morphological property that can be exploited to enhance the efficiency of an automatic sentiment classification system. The proposed method seeks to use different rooting techniques to reduce input feature words to their most basic roots, thus mapping a larger number of words to matching source roots. Sentiment-bearing roots, once found recurrently in a positive or negative context, can be used to classify many more words than the derivations themselves, allowing for classification of a broader feature set.

Two sets of sentiment-bearing words that are derived from the positive-sentiment root ن ج ح (nun-jim-ha', having to do with “success”) and the negative-sentiment root ق ت ل (qaf-ta-lam, having to do with “killing”), are shown in Table 2 (with their respective transliterations and translations). Various derivations are shown in matching positive and negative contexts, where the root word is the same, and the meaning of the sentence, likewise, retains the same sentiment orientation.

Because the task of sentiment analysis is not enclosed at the word-level, and because the surrounding words in a phrase may change the meaning of the phrase significantly as in the case of polarity incrementing or decrementing words (or even entirely, as in the case of negation words) a root-matching scheme on its own is not sufficient for consistently accurate sentiment classification. One common handling of such problems as negations in English is to consider all words between the negation and the next clause-level punctuation mark as negative (Pang et al., 2002; Sanjiv and Chen, 2001). In an Arabic context, a more flexible free word-ordering makes such a method difficult to consistently match, so the task would require a more elaborate handling scheme. Still, the initial root-matching task can be used to enhance results as a building block for an automatic Arabic sentiment analysis system.

### 3 Related Work

While there has been much work on sentiment analysis in English, few examples of work on the

Arabic Word	Positive Word Context Excerpt
انجح ānğħ “the most successful”	يمكن اعتبار [...] من انجح المديرين ymkn āṭbār [...] mn ānğħ ālmdrbyn “[...] can be considered one of the most successful coaches”
النجاحات ālnğāħāt “successes”	النجاحات العديدة التي حققها مع المنتخب ālnğāħāt ālḍydh ālty ħqğħā m' ālmnthb “the many success that [he] achieved with the team”
نجحت nğħt “succeeded”	نجحت [...] في تجديد عقود ابرز نجوم الفريق nğħt [...] fy tğdyd qwd ābrz nğwm ālfryq “[...] succeeded to renew the contracts of the most prominent team stars”
Arabic Word	Negative Word Context Excerpt
قتل qtl “were killed”	قتل ١٠٧ اشخاص في المواجهات qtl 107 āšħāṣ fy ālmwāğħāt “107 people were killed in clashes”
مقتل mqtl “the killing [of]”	كانت حصيلة اولية اشارت مساء الجمعة الي مقتل اثنين kānt ħṣylħ āwlyħ āšārt msā ālğmħ āly mqtł ātnyn “the initial toll on Friday evening indicated the killing of two”
قتلي qtly “victims”	اعلن [...] مسؤوليته عن هجومين [...] اوقعا خمسة قتلي āḍn [...] msūwlyth n ħğwmyn [...] āwqā ħmsh qtly “[he] claimed responsibility for two attacks leaving five victims”

Table 2: Two Sets of Sentiment-Bearing Words Derived from Common Roots, in Context. (Excerpts from the PATB Part 1 v 4.1 (Maamouri et al., 2010))

task for morphologically complex language such as Arabic are available, and possibly even more rare are data sets and corpora suitable for work on Arabic sentiment classification tasks.

Pang et al. (2002) tackled the classic problem of positive and negative two-class sentiment classification of English movie reviews from the Internet Movie Database (IMDB) corpus, highlighting the effectiveness of machine-learning techniques for sentiment classification, and paving the way for further research to enhance the efficiency of such automatic classification systems. With respect to the varied levels of granularity used (term, phrase, sentence, and document) in the classification task, individual and process-oriented approaches have been addressed, where information acquired from one level of analysis can be passed on to the next level, as observed by Turney and Littman (2003) and Dave et al. (2003). At the sentence-level, the work of Kim and Hovy (2004) addresses the topic of detecting sentiment towards a specific topic.

For feature selection and optimization, Yu et Hatzivassiloglou (2003) use N-gram based features and a polarity lexicon at the sentence level to determine subjectivity of sentences on the Wall Street Journal (WSJ) corpus, while Bruce and Wiebe (1999) use the same corpus, but employ additional lexical, part-of-speech (POS), and structural features. As a more profound paradigm shift, recent research has shifted from keyword and lexical-based approaches to concept-based sentiment analysis approaches, where semantic networks and entity ontologies are employed to achieve a more semantically-oriented “understanding” of text (Cambria et al., 2013; Grassi et al., 2011; Olsher, 2012).

With respect to the task of Arabic subjectivity and sentiment analysis in specific, the work of Abdul-Mageed et al. (2011) addresses the task in Modern Standard Arabic (MSA), where a manually-annotated corpus of MSA is presented from Part 1 v 3.0 of the Penn Arabic Treebank (PATB) (Maamouri et al., 2004), in addition to a wide-scale polarity lexicon tailored to the newswire domain under analysis. By using various stemming and lemmatization settings with a rich feature-set under an SVM classifier, it is shown that taking language-specific morphological features and traits into consideration for complex languages such as Arabic results in significant improvements in performance, achieving

test results of 71.54% F (16.44% higher than the baseline) for subjectivity detection, and 95.52% F (37.14% higher than the baseline) for the sentiment analysis task using a newswire domain-specific lexicon, as compared to 57.84% F in development without the lexicon.

In the domain of positive and negative movie reviews, Rushdi-Saleh et al. (2011) present the Opinion Corpus for Arabic (OCA) movie review corpus, compiled from various Arabic web pages. Classification was performed using both Naive-Bayes and SVM classifiers, with combinations of N-grams, stemming, and stop-word removal pre-processing, and achieved a best result of 90% accuracy under SVM, as compared to a similar classification task in English with the Pang et al. (2002) IMDB corpus, which obtained 85.35% accuracy with various N-gram models.

Abbassi et al. (2008) explore the task of feature selection for the opinion classification task, using an Entropy Weighted Genetic Algorithm (EWGA), which incorporates both syntactic and stylistic features and the information gain heuristic to classify text on the document level. An accuracy of 93.6% is reported on a compiled Middle-Eastern web forum dataset. Other problems of sentiment analysis in informal and dialectal Arabic are also addressed by Albraheem and Al-Khalifa (2012) and Shoukry and Rafea (2012), for a more specific approach to the classification problem, tailored to a regional, social-media setting.

As compared to the discussed work on the sentiment analysis task in Arabic, the proposed root-based technique employs the commonalities between words of the same root to map sets of words to the same base meaning. Rather than taking a domain-specific approach to the problem, the proposed technique is tested on two corpora from very different domains: the PATB newswire corpus annotated by Abdul-mageed et al. (2011), and the OCA movie review corpus by Rushdi-Saled et al. (2011), with focus on language characteristics to enhance classification results.

## 4 Proposed Algorithm

The following section presents a detailed description of the datasets and rooting libraries used for experimentation, the various experimentation settings undergone, and the proposed classification method applied for the task sentiment analysis on the two studied domains.

## 4.1 Datasets

The proposed sentiment classification method was conducted on two different datasets, to test the root-based approach on a generic level, unconstrained by the domain of the data itself.

### 4.1.1 Penn Arabic Treebank (PATB Part 1 v 4.1) Newswire Corpus

The first corpus used pertains to the tokenized newswire-domain text from the latest version of the PATB, Part 1 v 4.1 (Maamouri et al., 2010). The corpus consists of 734 newswire stories from the Agence France Presse (AFP) with various tags attached to each token, including part-of-speech information, morphology, English gloss, treebank annotation, and vocalization.

For the purposes of the sentiment analysis task, the applied section of the dataset comes from the compiled corpus of Abdul-Mageed et al. (2011), where the first 2855 sentences (comprising 54.5% of the Part 1 v 3.0 dataset<sup>1</sup>, in 400 documents) were each manually annotated into one of four labels (Abdul-Mageed and Diab, 2011): objective (OBJ), subjective positive (S-POS), subjective negative (S-NEG), and subjective neutral (S-NEUT), depending on whether the information being conveyed in the sentence was to objectively inform, or offer a subjective sense (Wiebe et al., 1999). The number of sentences with each of the four respective tags are shown in Table 3.

Tag Class	Number of Sentences
OBJ	1281
S-POS	491
S-NEG	689
S-NEUT	394

Table 3: Distribution of Sentiment Classes in the Manually-Tagged Portion of the PATB Corpus (Abdul-Mageed et al., 2011).

### 4.1.2 Opinion Corpus for Arabic (OCA) Movie Review Corpus

For another perspective for testing the proposed root-based method, the distinctly subjective OCA corpus (Rushdi-Saleh et al., 2011) was also experimented with. The corpus is comprised of

<sup>1</sup>The differences between the two versions of the PATB Part 1 lie in improvements to the organization of the data, and updates to certain aspects of the annotation (Maamouri et al., 2010).

500 movie reviews (250 positive and 250 negative) which were collected from 15 Arabic web pages, after which a series of spelling correction, tokenization, basic stop-word and special character removal, and stemming processes were performed. In addition, normalization of the rating schemes used for each site was conducted to appropriately partition the reviews into positive and negative categories, and prepare the text for the classification task (Rushdi-Saleh et al., 2011).

## 4.2 Rooting Libraries

Three different rooting libraries were applied to derive the roots of each of the input words in the classification examples, each applying a slightly different approach to the complex Arabic rooting problem.

### 4.2.1 Khoja Arabic Stemmer

The Khoja Arabic Stemmer (Khoja and Garside, 1999) is a fast Arabic stemmer that works by removing the longest prefix and suffix present in the input word and then matching the rest of the word with known verb and noun patterns using a root library. The stemmer attempts to take into account the unavoidable irregularities in the language in order to extract the correct root from words that do not follow the general rules. The Khoja stemmer has been used in various Arabic natural language processing tasks, and has been noted to produce good improvements to various natural language tasks, despite many tagging errors (Larkey and Connell, 2001).

### 4.2.2 Information Science Research Institute (ISRI) Arabic Stemmer

The Information Science Research Institute's (ISRI) stemmer (Taghva et al., 2005) uses a similar approach to word rooting as the Khoja stemmer, but does *not* employ a root dictionary for lookup. Additionally, if a word cannot be rooted, the ISRI stemmer normalizes the word and returns a normalized form (for example, removing certain determinants and end patterns) instead of leaving the word unchanged. The ISRI stemmer has been shown to give good improvements to language tasks such as document clustering, as opposed to a non-stemmed approach (Bsoul and Mohd, 2011).

### 4.2.3 Tashaphyne Light Arabic Stemmer

The Tashaphyne Light Arabic Stemmer (Tashaphyne, 2010) works by first normalizing words in



preparation for the “search and index” tasks required for stemming, including removing diacritics and elongation from input words. Next, segmentation and stemming of the input is performed using a default Arabic affix lookup list, allowing for various levels of stemming and rooting (Tashaphyne, 2010).

### 4.3 Experimental Setup

Two different sets of experiments were conducted to test the effect of the root-based method for the sentiment analysis task, varying somewhat for the different corpora under analysis.

For the PATB corpus, only the subjective data was taken into consideration for the two-class positive and negative sentiment classification problem (the 1180 sentences in total: 491 S-POS and 689 S-NEG sentences). The task was conducted at the sentence-level, with 5-fold cross-validation splits across the dataset (Abdul-Mageed et al., 2011).

For the OCA corpus, with already-defined opinions in the form of movie reviews, the entire dataset was used for classification. The task was conducted at the document-level (with each document composed of sets of sentences, ranging from an average of 13 sentences in the positive review sets, and 20 sentences in the negative reviews), with 10-fold cross-validation splits (Rushdi-Saleh et al., 2011).

Each of the corpora was tested using the basic words as features to the classifier, then by iteratively adding roots using each of the three rooting libraries. After experimentation with parameters, the classification task was performed using a linear kernel (Abdul-Mageed et al., 2011) under the SVM<sup>light</sup> classifier (Joachims, 2008). Precision and recall values are reported for the average of the K-fold runs (5 folds for the first corpus, and 10 folds for the second), along with F-measure (F1) and accuracy results for each respective experiment.

### 4.4 Pre-processing

As pre-processing to prepare the text for the classification and analysis tasks, the already undiacritized corpus sentences were tokenized from the set of documents into word sets, and testing with stop-word removal (the removal of commonly used words) was done to filter out words that could be unnecessary for the task of opinion classification.

### 4.5 Feature Sets

For each document, the basic unigrams (individual words) composing the document were used as initial input features to the SVM, after which the three rooting libraries (Khoja, ISRI, and Tashaphyne) were then iteratively used to derive the roots of each of the input word features. Finally, the resultant roots were then added as additional features (along with the unigrams) to each of the examples. Binary presence vectors were used to indicate the existence of a feature (Abdul-Mageed et al., 2011). For the purposes of exploring the effect of adding the root-based features, only independent unigrams and unigrams with roots were experimented with for the basic evaluation task.

## 5 Results and Evaluation

The results of the sentiment analysis tasks on the two datasets are illustrated in Table 4 for the PATB corpus, and Table 5 for the OCA corpus, detailing the task statistics. The basic results using a standard *unigram* feature (the encountered word itself) are depicted initially, along with a *baseline* result (in F-measure and accuracy, as available for the two datasets, respectively) without the use of root features, as presented by previous work with the datasets (Abdul-Mageed et al., 2011; Rushdi-Saleh et al., 2011).

With respect to the sentiment analysis task on the PATB newswire-domain corpus shown in Table 4, all three of the individual rooting libraries resulted in improvements to the initial unigram results. The largest observable improvement to all measures reported came from the Khoja stemmer, with a 4.9% increase in F-measure, and a 4.7% increase in accuracy as compared to the unigram result. Also, a 3.4% increase in F-measure is observed from the sentiment baseline of 57.8% (previously achieved on the dataset using various morphological features, without a domain-specific lexicon (Abdul-Mageed et al., 2011)).

For the OCA movie-domain corpus shown in Table 5, slight improvements can be seen by adding root features to the unigram classifier input, particularly with the Tashaphyne rooting library. An increase of 3.2% accuracy after the addition of root features is observed from the baseline accuracy of 90.0% (Rushdi-Saleh et al., 2011).

While an increase in overall accuracy and F-measure is notable in the task of basic two-class opinion classification, two main points are of im-

	Precision	Recall	F1	Accuracy
<b>Unigrams (No Roots)</b>	58.1	58.1	56.3	63.8
<b>+ Khoja Roots</b>	<b>63.8</b>	<b>61.9</b>	<b>61.2</b>	<b>68.5</b>
<b>+ ISRI Roots</b>	61.5	60.4	59.3	66.8
<b>+ Tashaphyne Roots</b>	61.7	58.8	58.8	67.0
Baseline	57.8			

Table 4: PATB Sentiment Classification Results for the Proposed Method under Three Rooting Libraries. (Baseline: Abdul-Mageed et al. (2011))

	Precision	Recall	F1	Accuracy
<b>Unigrams (No Roots)</b>	90.0	95.2	92.8	92.6
<b>+ Khoja Roots</b>	90.7	94.4	92.3	92.2
<b>+ ISRI Roots</b>	90.5	95.6	92.8	92.6
<b>+ Tashaphyne Roots</b>	<b>91.1</b>	<b>96.0</b>	<b>93.4</b>	<b>93.2</b>
Baseline	90.0			

Table 5: OCA Sentiment Classification Results for the Proposed Method under Three Rooting Libraries. (Baseline: Rushdi-Saled et al. (2011))

portance: the nature of the words in the dataset under analysis, and the efficiency of the stemming systems themselves. The divergence in the most effective rooting library on each of the corpora can be attributed to various factors, including the style of writing used in the datasets, the formality of the text, and the existence of irregular words and words that cannot be rooted, depending on the accuracy and robustness of the employed stemming library.

The PATB news domain corpus, for example, is expected to have less opinion-bearing content than the OCA movie review corpus, due to the less subjective nature of the domain. The overall accuracy and F-measure results for the OCA movie corpus are thus significantly higher than those observed in the PATB corpus. Another difference between the corpora lies in the formality of the language employed: while the PATB corpus uses a strict Modern Standard Arabic (MSA), the use of slang and dialect-specific language is frequent in the OCA corpus. This type of varied language presents a layer of difficulty for sentiment classification in general, as well as for the rooting systems applied for language mapping.

Furthermore, with respect to the stemming tools themselves, the overall inaccuracy of current stemmers is another important consideration. The best-performing stemming libraries, Khoja and Tasha-

phyne, for each of the two domains, are those that employ some form of root-lookup dictionary in order to verify the correctness of the affixes and resultant roots generated. Another consideration is the limitation imposed by employing only unigrams enriched with the root features, while features beyond word level could be used to further predict sentiment patterns changes over a more complex language structure.

## 6 Conclusion

The composition scheme and complex morphology of Arabic make the task of root-extraction to normalize words to their basic functional units a very relevant one for various natural language processing tasks. With respect to the sentiment analysis tasks presented in this paper, some notable improvements to the classification performance when using various rooting libraries as input features can be observed, warranting further research on enhancements to existent rooting techniques and handling of the intricacies of the Arabic language structure to predict more sentence forms and correctly classify their polarity.

As detailed, the reasoning behind the root-based method and its enhancement of the sentiment classification task in the two explored domains relies on the semantic similarities between different word derivations, allowing for a broader map of

interconnections between words with similar polarity orientation to be created. Such valid interconnections between words also warrants the exploration of semantic expansion of words and their synonyms (Magdy et al., 2013), expanding the word map and serving to better connect and understand sentiment-bearing ideas and expressed opinions.

By applying various rooting schemes at different granularities in two separate domains, it is also shown that word roots can serve to enhance the sentiment analysis task results on a more generic level, instead of using a domain-specific approach that may not always be applicable. Thus, using root derivation techniques such as that presented for Arabic sentiment analysis in particular are applicable and valid to help enhance the performance of natural language processing tasks on morphologically rich and complex languages.

## References

- A. Abbasi, H. Chen, and A. Salem. 2008. Sentiment analysis in multiple languages: feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):12:1–12:34, June.
- M. Abdul-Mageed and M. Diab. 2011. Subjectivity and sentiment annotation of modern standard arabic newswire. In *Proceedings of the 5th Linguistic Annotation Workshop, LAW V '11*, pages 110–118, Stroudsburg, PA, USA. Association for Computational Linguistics.
- M. Abdul-Mageed, M. Diab, and M. Korayem. 2011. Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, pages 587–591, Stroudsburg, PA, USA. Association for Computational Linguistics.
- L. Albraheem and H. Al-Khalifa. 2012. Exploring the problems of sentiment analysis in informal arabic. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications; Services, IIWAS '12*, pages 415–418, New York, NY, USA. ACM.
- R. Bruce and J. Wiebe. 1999. Recognizing subjectivity: a case study in manual tagging. *Nat. Lang. Eng.*, 5(2):187–205, June.
- Q. Bsoul and M. Mohd. 2011. Effect of isri stemming on similarity measure for arabic document clustering. In *Information Retrieval Technology*, volume 7097 of *Lecture Notes in Computer Science*, pages 584–593. Springer Berlin Heidelberg.
- E. Cambria, B. Schuller, Y. Xia, and C. Havasi. 2013. New avenues in opinion mining and sentiment analysis. *Intelligent Systems, IEEE*, 28(2):15–21.
- K. Dave, S. Lawrence, and D. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web, WWW '03*, pages 519–528, New York, NY, USA. ACM.
- M. Grassi, E. Cambria, A. Hussain, and F. Piazza. 2011. Sentic web: A new paradigm for managing social media affective information. *Cognitive Computation*, 3(3):480–489.
- N. Habash. 2008. *Introduction to Arabic Natural Language Processing*. Synthesis lectures on human language technologies. Morgan & Claypool Publishers.
- T. Joachims. 2008. Svm-light: Support vector machine, <http://svmlight.joachims.org/>.
- S. Khoja and R. Garside. 1999. Stemming arabic text (tech. rep.). Computer Department, Lancaster University, Lancaster.
- S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics, COLING '04*, Stroudsburg, PA, USA. Association for Computational Linguistics.
- L. Larkey and M. Connell. 2001. Arabic information retrieval at umass in trec-10. University of Massachusetts.
- B. Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
- M. Maamouri, A. Bies, and W. Mekki. 2004. The penn arabic treebank: building a large-scale annotated arabic corpus. pages 102–109.
- M. Maamouri, A. Bies, S. Kulick, F. Gaddeche, W. Mekki, S. Krouna, B. Bouziri, and W. Zaghouni. 2010. Arabic treebank: Part 1 v 4.1.
- A. Magdy, M. Kholief, and Y. El-Sonbaty. 2013. Qasim: Arabic language question answer selection in machines. *Conference and Labs of the Evaluation Forum*.
- D. Olsher. 2012. Full spectrum opinion mining: integrating domain, syntactic and lexical knowledge. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 693–700.
- B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on empirical methods in natural language processing - Volume 10, EMNLP '02*, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

- M. Rushdi-Saleh, M. Martin-Valdivia, L. Urena-Lopez, and J. Perea-Ortega. 2011. Oca: Opinion corpus for arabic. *J. Am. Soc. Inf. Sci. Technol.*, 62(10):2045–2054, October.
- K. Ryding. 2006. *A Reference Grammar of Modern Standard Arabic*. Reference Grammars. Cambridge University Press, New York.
- D. Sanjiv and M. Chen. 2001. Yahoo! for amazon: extraction market sentiment from stock message boards. In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*.
- A. Shoukry and A. Rafea. 2012. Sentence-level arabic sentiment analysis. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 546–550.
- StudyQuran. 2004. Project root list online. 2004. <http://www.studyquran.co.uk/prlonline.htm>.
- K. Taghva, R. Elkhoury, and J. Coombs. 2005. Arabic stemming without a root dictionary. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume 1 - Volume 01, ITCC '05*, pages 152–157, Washington, DC, USA. IEEE Computer Society.
- Tashaphyne. 2010. Arabic light stemmer, 0.2. 2010. <http://tashaphyne.sourceforge.net/>.
- P. Turney and M. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, October.
- J. Wiebe, R. Bruce, and T. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99*, pages 246–253, Stroudsburg, PA, USA. Association for Computational Linguistics.
- H. Yu and V. Hatzivassiloglou. 2003. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing, EMNLP '03*, pages 129–136, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Topical Key Concept Extraction from Folksonomy

Han Xue<sup>1,2</sup>, Bing Qin<sup>1\*</sup>, Ting Liu<sup>1</sup>, Chao Xiang<sup>1</sup>

<sup>1</sup>Research Center for Social Computing and Information Retrieval

Harbin Institute of Technology, Harbin, China

{hxue, bqin, tliu}@ir.hit.edu.cn, Cloudaice@gmail.com

<sup>2</sup>Harbin Engineering University, Harbin, China

## Abstract

Concept extraction is a primary subtask of ontology construction. It is difficult to extract new concepts from traditional text corpus. Moreover, building a single ontology for multiple-topic corpus may lead to misconception. To deal with these problems, this paper proposes a novel framework to extract topical key concepts from folksonomy. Folksonomy is a valuable data source due to real-time update and rich user-generated contents. We first identify topics from folksonomy using topic models. Next the tags are ranked according to their importance for a certain topic by applying topic-specific random walk methods. The top-ranking tags are extracted as topical key concepts. Especially, a novel link weight function which combines the local structure information and global semantic similarity is proposed in importance score propagation. From the perspectives of qualitative and quantitative investigation, our method is feasible and effective.

## 1 Introduction

Ontology can be seen as an organized structure of concepts according to their relations (Cui et al., 2009). Therefore, concept extraction is an important subtask of ontology construction. Existing works mainly focus on extracting concepts from text corpus (Buitelaar et al., 2005). However, it is difficult to find text corpus that accurately characterize a highly focused, even fast-changing topic (Liu et al., 2012) of the domain. For instance, it is easier to find text corpus for a common topic of movie such as “comedy”, but it

is much more difficult to find one for a specific topic such as “cult”. Since “cult” movies often do not follow traditional standards of mainstream movies. Moreover, it is not easy for them to find a formal definition and description in text corpus. However, we can more easily find some tags (arbitrary words assigned by people to the resources of interest) to describe this kind of movie, such as cult, non-mainstream, small budgets and so on. Motivated by the fact that social tags give us flexibility and ease to describe a topic, we try to use folksonomy (Trant, 2009) as a new data source. The word ‘folksonomy’ is a blend of the words ‘folk’ and ‘taxonomy’. It is the achievement of collective wisdom derived from the practice of collaboratively creating tags to annotate and categorize web resource.



Figure 1. A folksonomy example

Take Douban.com for example, which is a Chinese SNS website allowing registered users to record information and create tags related to

\*Correspondence author

their interested resources, such as film, books, music and recent activities. As shown in Fig. 1, in the folksonomy-driven web site 豆瓣电影网站‘Douban.com Movie<sup>1</sup>’, the resource 致我们终将逝去的青春‘So Young’ is annotated with a set of tags including 青春‘youth’, 爱情‘romance’, and 成长‘growth’ ordered by the frequency of use which update automatically.

Compared with traditional text corpus, folksonomy can overcome the knowledge acquisition bottleneck. It is superior to text corpus in three aspects. (1) tag is more free and easier to characterize a highly focused, even fast-changing topic; (2) tag as a candidate concept has been extracted by collective wisdom, which avoids a series of natural language processing tasks applied to text corpus such as word segmentation, part of speech tagging, and syntactic parsing and so on; (3) the associated relationships among resources, tags and users through tagging provide a large amount of potentially valuable semantic information for mining. However, folksonomy also has two disadvantages, such as ambiguity and lack of hierarchy. To avoid misconception, we think of building multiple topic-specific ontologies instead of a single one.

In this paper, we propose to automatically extract topical key concepts from folksonomy. The topical key concept should be abstract, representative of the corresponding topic. It should contain common features that can be inherited by other non-core relevant concepts under the same topic. For example, the topical key concepts in the field of movie may be comedy, biography and action and so on. To extract the topical key concepts, we learn the topic distribution of the tags by applying LDA (Latent Dirichlet Allocation) (Blei et al., 2003) at first. After that, the tags are ranked on the basis of the importance scores for a certain topic by a variant of topic-specific PageRank (Page et al., 1999). Specially, the novel contribution of the variant is a new link weight function in importance propagation, which combines the local similarity (defined as co-occurrence of tags in a same resource assigned to the given topic) with the global similarity (defined as cosine similarity of two tags over all the topic dimensions in the whole collection considered). Then, the top-ranking tags that best represent the corresponding topic are extracted as topical key concepts.

In view of limited Chinese corpus and complex Chinese syntax for ontology construction, we tried on Chinese folksonomy data. Experiments on movie data from Douban.com show that new link weight function can largely help boost the performance. To the best of our knowledge, our work is the first to study how to extract topical key concepts from folksonomy in the field of Chinese ontology construction. We perform a thorough analysis of the proposed method, which can be useful for future work in this direction.

Although our goal is to build Chinese ontology based on the topical key concepts from this work, our method can be widely used in many other tasks such as information navigation and recommendation system. Furthermore, our method is unsupervised and language independent, which is applicable in the web era with enormous information.

The rest of the paper is organized as follows. Section 2 reviews some related works; Section 3 describes our proposed method; Section 4 presents our experiments and resultant analysis; and Section 5 draws the conclusions and directions for the future work.

## 2 Related Work

Many efforts have been made to extract the key concepts for ontology construction. These methods can be divided into two categories according to topic-sensitive or not.

**Topic-free** Some key concepts of famous ontologies are usually defined by linguists or domain experts. The suggested upper merged ontology (SUMO) is such a kind of ontologies. The expert-based methods are accurate and standard. However, to tackle the time-consuming and laborious problems, efforts are also made to use semi-automatic and automatic methods.

Among semi-automatic methods, rule-based methods are known for high accuracy if the patterns are carefully chosen according to morphological structure or special format of corpus (Nakayama et al., 2008), either manually or via automatic bootstrapping (Hearst, 1992). However, the methods suffer from sparse coverage of patterns in a given corpus.

Some researchers try to map the words to a thesaurus or an existed ontology (WordNet or Wikipedia) automatically so as to get key concepts (Angeletou et al., 2008). The coverage and openness of existed ontologies seriously limit the scope of these works. Simple statistical methods

---

<sup>1</sup> <http://movie.douban.com>

such as TF-IDF weighting (Hulth, 2003) are not feasible for folksonomy since short text snippets only. Graph-based ranking methods are the state of the art. They are superior to the statistic-based methods because of considering structure information between words. Mihalcea and Tara (2004) propose to use TextRank, a modified PageRank algorithm to extract key concepts from text. But TextRank only maintain a single importance score for each word. Hotho et al. (2006) propose a graph-based ranking algorithm for folksonomy, named FolkRank. They convert triadic hypergraph in folksonomy into an undirected tripartite graph. But we consider that the tripartite graph may include much noise for key concept extraction.

As a word usually spans multiple topics, the importance of the word with respect to different topics would be different. It seems that the previous works mentioned above may lead to misconception by mixing different topics together.

**Topic-sensitive** In order to overcome misconceptions, the topic models and other clustering models such as DA (Deterministic annealing) (Zhou et al., 2007) are used to derive topical key concepts from corpus based on word occurrence information. These clustering models usually regard corpus as a bag of words. They can find the topic or the leading word in each cluster, but cannot distinguish concrete entity well.

It is intuitive to consider topic information in graph-based ranking methods for topical key concept extraction.

Haveliwala (2002) proposes topic-sensitive PageRank (TSPR) to get a set of PageRank vectors biased a set of representative topics and generate more accurate rankings than a single PageRank vector. Nie et al. (2006) propose a topical link analysis model (TLA) to affect the importance propagation. However, the topics in TSPR and TLA are both from ODP (Open Directory Project)<sup>2</sup> extracted manually. Based on their works, Jin et al. (2011) implement a topic-sensitive tag ranking (TSTR) approach in folksonomy automatically through LDA. TSTR performs better than TSPR and TLA because topics extracted by LDA are more conformed to the actual situation than the topics of ODP. They pay more attention to the effect of the transfer action probability on the importance score of tags which benefit us to know the propagation process. It seems that mixing together the random

walks of all the topics in one graph may cause noise compared to independent topic graphs.

Liu et al. (2010) decompose a traditional random walk into multiple random walks specific to various topics, named Topical PageRank (TPR). The novel contribution is the study on topic-specific preference value setting. And then, Zhao et al. (2011) argue that context-free propagation may cause the importance scores to be off-topic. They model the score propagation with topic context when setting the link weights and then denote this context-sensitive topical PageRank as cTPR. Enlightened by TPR and cTPR, we further propose a new link weight function to express the semantic similarity between two tags of folksonomy. The novel link weight function combines the local similarity (defined as co-occurrence of tags in a same resource assigned to the given topic) with the global similarity (defined as cosine similarity of two tags over all the topic dimensions in the whole collection considered).

### 3 Method

In this section, we will introduce our method. We firstly give some definitions and then overview our method, and finally introduce topic identification and tag ranking in detail.

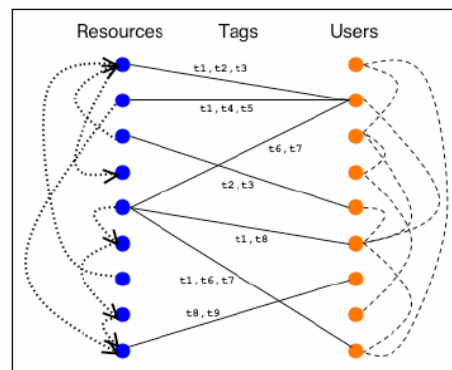


Figure 2. A conceptual model of folksonomy

From the Fig. 2 (Marlow, 2006), we can see a conceptual model of social tagging behavior in folksonomy. It consists of users  $u \in U$ , tags  $w \in V$ , and resources  $s \in S$ . The conceptual model illustrates visually the implicit association among resources, tags and users joined with straight lines and dashed lines. Folksonomy is composed of  $\langle U, V, S \rangle$  triples. For simplicity, we only regard a resource  $s \in S$  as a document which includes a set of tags assigned by all the users of collection  $U$ . Moreover, we suppose that there is

<sup>2</sup> <http://www.dmoz.org/>

a set of topics  $Z$  over the resource collection  $S$ . Hence, we extract topics from  $S$ .

The basic idea of our method is to incorporate topic distribution into importance score propagation of tags when setting the link weight as well as the preference value. Especially, the link weight function considers both the local and global similarity with respect to different topics.

First of all, we identify topics from folksonomy resource collection  $S$  using topic models (Section 3.1). Next for each topic, we build the tag graph and use the random walk techniques to measure the tag importance. Based on the importance scores, we extract the top-ranking tags as the topical key concepts (Section 3.2).

### 3.1 Topic Identification

Latent Dirichlet Allocation (LDA) is a typical representative of topic models. In LDA, each word  $w$  in a document  $d$  is regarded to be generated by first sampling a topic  $z$  from  $d$ 's topic distribution  $\theta$ , and then sampling a word from the distribution over words  $\Phi$  that characterizes topic  $z$ .  $\theta$  and  $\Phi$  are drawn from conjugate Dirichlet priors  $\alpha$  and  $\beta$ , separately.

We use resource set  $S$  represented by a set of tag as our input file to run LDA. After the parameters converge by Gibbs sampling, we mainly use two of these output files in this paper. One is model-final.phi, which is a  $|Z|*|V|$  matrix about  $\Phi$ , whose element is the probability of tag  $w_i$  conditional on topic  $z_j$  i.e.  $P(w_i | z_j)$ . The other is model-final.tassign, which is a  $|S|*|V|$  matrix, where each row of data stands for a resource  $s$  followed by a set of elements, and each element consists of a tag and a topic which the tag most likely to be assigned to.

Through LDA, we can obtain the topic distribution of each tag  $w_i \in V$  by Eq. 1, namely  $P(z | w_i)$  for given topic  $z \in Z$ ,

$$P(z | w_i) = \frac{P(z)P(w_i | z)}{\sum_{z'} P(z')P(w_i | z')} \quad (1)$$

where  $P(w_i | z)$  can be found in the model-final.phi directly, and  $P(z)$  is calculated by Eq. 2.

$$P(z) = \frac{C(z)}{\sum_{z'} C(z')} \quad (2)$$

In which,  $C(z)$  is calculated as the number of times topic  $z$  appears in the model-final.tassign.

Obviously,  $\sum_{z'} C(z')$  is calculated as the number of times all the topics appear in the model-final.tassign. Then, we can calculate the local and global similarity between two tags using Eq. 3 and Eq. 4.

$$Local_s(w_j, w_i) = \frac{C_{w_j, w_i, z}^S}{C_{w_j, w_i}^S} \quad (3)$$

$$Global_s(w_j, w_i) = \frac{\sum_z^S p(z | w_j)p(z | w_i)}{\sqrt{\sum_z^S p(z | w_j)^2 \sum_z^S p(z | w_i)^2}} \quad (4)$$

Among them,  $Local_s(w_j, w_i)$  stands for the local semantic similarity between tag  $w_i$  and  $w_j$ . In Eq.

3,  $C_{w_j, w_i, z}^S$  counts the number of co-occurrences of tag  $w_i$  and  $w_j$  in a same resource of  $S$  assigned to the topic  $z$ .  $C_{w_j, w_i}^S$  counts the number of co-occurrences of tag  $w_i$  and  $w_j$  in a same resource of  $S$ . We can get them from statistical calculation of the model-final.tassign.  $Global_s(w_j, w_i)$  stands for the global semantic similarity between tag  $w_i$  and  $w_j$ . From the whole resource collection  $S$ , we can get the cosine similarity between tag  $w_i$  and  $w_j$  over all the topic dimensions by plugging Eq. 1 into Eq. 4.

### 3.2 Tag Ranking

After topic identification, we perform topical key concept extraction followed by two steps, namely, tag graph construction and tag ranking.

Above all, some formal notations are given. We denote  $G = (V, E)$  as the graph composed of tags, with vertex set  $V = \{w_1, w_2, \dots, w_N\}$  and link set  $(w_i, w_j) \in E$  if there is a link from node  $w_i$  to  $w_j$ . In a tag graph, each vertex represents a tag, and each link indicates the correlation between every two tags. We denote the weight of the link  $(w_i, w_j)$  as  $e(w_i, w_j)$ , and the out-degree of vertex  $w_i$  as  $O(w_i) = \sum_{j: w_i \rightarrow w_j} e(w_i, w_j)$ .

PageRank assigns global importance scores to vertices using link information. In PageRank, the score  $R(w_i)$  of the word  $w_i$  is defined as

$$R(w_i) = \lambda \sum_{j: w_j \rightarrow w_i} \frac{e(w_j, w_i)}{O(w_j)} R(w_j) + (1 - \lambda) \frac{1}{|V|} \quad (5)$$



where damping factor  $\lambda$  ranges from 0 to 1, and  $|V|$  is the number of vertices. The damping factor indicates that each vertex has a probability of  $(1-\lambda)$  to perform a random jump to another vertex within this graph while has a probability of  $\lambda$  to follow the out-degree link. The PageRank importance scores are obtained by running Eq. 5 iteratively until convergence. The second term in Eq. 5 can be regarded as a smoothing factor to make the graph fulfill the property of being aperiodic and irreducible, so as to guarantee that PageRank converges to a unique stationary distribution.

In fact, the second term of PageRank in Eq. 5 can be set to be non-uniformed. The idea of Topical PageRank (TPR) is to run Biased PageRank for each topic separately. Formally, in the PageRank of a specific topic  $z$ , they assign a topic-specific preference value  $\text{Pr}_z(w)$  to each word  $w$  as its random jump probability with  $\sum_{w \in V} \text{Pr}_z(w) = 1$ . For topic  $z$ , the topic-specific PageRank importance scores are defined as follows,

$$R_z(w_i) = \lambda \sum_{j:w_j \rightarrow w_i} \frac{e(w_j, w_i)}{O(w_j)} R_z(w_j) + (1-\lambda) \text{Pr}_z(w_i) \quad (6)$$

However, TPR ignores the topical context in the link weight settings; the link weight  $e(w_j, w_i)$  in Eq.6 is calculated as the number of co-occurrences of two words within a certain window size. Zhao et al. (2011) propose to use a topical context-sensitive PageRank method (cTPR). Formally, they have

$$R_z(w_i) = \lambda \sum_{j:w_j \rightarrow w_i} \frac{e_z(w_j, w_i)}{O_z(w_j)} R_z(w_j) + (1-\lambda) \text{Pr}_z(w_i) \quad (7)$$

where they calculate the propagation from  $w_j$  to  $w_i$  in the context of the topic  $z$ , namely, the link weight  $e_z(w_j, w_i)$  from  $w_j$  to  $w_i$  is parameterized by  $z$ . In Eq.7, the link weight between two words is calculated as the number of co-occurrences of the two words in tweets assigned to the topic  $z$ .

However, we believe that the link weight only contains the co-occurrence information is not enough. On the basis of cTPR, we further propose a new link weight function (Eq.8).

$$e_z(w_j, w_i) = \text{Local}_s(w_j, w_i) \rho + (1-\rho) \text{Global}_s(w_j, w_i) \quad (8)$$

The weight factor  $\rho$  controls the proportion of the local structure information (Eq.3) and global semantic similarity (Eq.4) in Eq.8. Through the new link weight, the propagation of our method not only reflects the specific local co-occurrence information of two tags in a single resource, but also reflects the non-specific global semantic similarity of two tags in the whole resource set.

In our tag ranking method, we obtain the importance scores for each tag in different topics by Eq.7. In which,  $e_z(w_j, w_i)$  is calculated by Eq.8, and  $O_z(w_j) = \sum_{i:w_j \rightarrow w_i} e(w_j, w_i)$ . The topic-specific preference value  $\text{Pr}_z(w_i) = P(z | w_i)$  is calculated as Eq.1, which is the best one among the three choices discussed by Liu et al. (2010).

## 4 Experiments

### 4.1 Dataset

Our evaluation dataset is crawled from Douban.com Movie, which is a popular Chinese Social Networking Service (SNS) website allowing registered users to create content related to movies. The dataset contains top 250 movies with 1760 tags assigned by users up to June 2012. After removing stop words and noises, we prepare 1737 tags corresponding to 249 movies for LDA. Empirically, we set the number of topics to 40 and ran LDA with 1000 iterations of Gibbs sampling.

We further select two baseline methods that most similar to ours, i.e., TPR and cTPR. All of them are iterative algorithms. We terminate the algorithms when the number of iterations reaches 100 or the difference of importance scores about each vertex between two neighbor iterations is less than 0.000001.

There are three parameters in our method that may affect the performance of the topical key concept extraction including (1) damping factor  $\lambda$  that reconciles the influence of adjacent nodes' importance (the first item in Eq. 7) and preference value (the second item in Eq. 7) to the modified PageRank importance of our method; (2) weight factor  $\rho$  that controls the proportion of the local structure information (Eq. 3) and global semantic similarity (Eq. 4) on two tags; (3) threshold  $Q$ ; If the global semantic similarity between two tags is less than  $Q$ , we will remove the link between them. We separately set parameters  $\lambda$ ,  $\rho$  and  $Q$  from 0.1 to 0.9 with a step size of 0.1, and then each parameter has 9 candidate values. Finally, 729 experiment results of the

baseline and our method based on permutation and combination of the three parameters are presented.

## 4.2 Gold Standard Annotation

We construct the evaluation standard by pooling (Voorhees et al., 2005) method. The reason lies in two aspects. One is that there is no existing gold standard for topical key concept extraction from folksonomy, and the other is that it is impossible to determine all the topics and key concepts manually. We randomly mix 729 results from TPR, cTPR and our method, and then ask two judges to score as 1 (relevant, abstract and representative) or 0 (irrelevant or too specific). Only if the two judges score 1 for the same tag, the tag will be determined as correct topical key concept. Otherwise, the tag will be determined as wrong.

## 4.3 Evaluation Metrics

The traditional evaluation metrics represented as follows,

$$\begin{aligned} P &= \frac{C_{correct}}{C_{extract}}, \\ R &= \frac{C_{correct}}{C_{standard}}, \\ F &= \frac{2PR}{P+R} \quad (9) \end{aligned}$$

where  $C_{correct}$  denotes the number of correct topical key concepts extracted by a method,  $C_{extract}$  denotes the number of automatically extracted topical key concepts by a method, and  $C_{standard}$  denotes the total number of topical key concepts referenced by gold standard. Three of them are averaged on all the topics.

In addition to the traditional metrics precision/recall/F-measure, we use another two metrics to take the order into account.

One metric is mean average precision (MAP). MAP is desirable to measure the overall performance of topical key concept ranking,

$$MAP = \frac{1}{|Z|} \sum_{z \in Z} \frac{1}{N_z} \sum_{j=1}^{|M_z|} \frac{N_{M_z, z, j}}{j} I(score(M_{z, j}) \geq 1) \quad (10)$$

where  $I(S)$  denotes an indicator function which returns 1 when  $S$  is true and 0 otherwise,  $N_{M_z, z, j}$  denotes the number of correct key concepts among the top  $j$  candidates returned by

method  $M$  for topic  $z$ , and  $N_z$  denotes the total number of correct key concepts of topic  $z$  referenced by the gold standard.

The other metric is mean reciprocal rank (MRR) (Voorhees, 1999) which is used to evaluate how the first correct topical key concept for each topic is ranked. For a topic  $z$ ,  $rank_z$  is denoted as the rank of the first correct topical key concept with all extracted candidates, MRR is defined as follows,

$$MRR = \frac{1}{|Z|} \sum_{z \in Z} \frac{1}{rank_z} \quad (11)$$

where  $Z$  is the topic set for topical key concept extraction.

## 4.4 Quantitative Evaluation

As for the same folksonomy dataset from Douban.com Movie, we realize the baseline methods, i.e., TPR and cTPR. The TPR calculates the co-occurrence number of two tags in a same resource as the link weight, namely  $e(w_j, w_i) = C_{w_j, w_i}^S$ .

In cTPR, the link weight is calculated as the co-occurrence number of two tags in a same resource assigned to the same given topic, namely  $e_z(w_j, w_i) = C_{w_j, w_i, z}^S$ .

The grid-search algorithm is applied to obtain the optimal parameter combination from 729 candidates. Through an exhaustive combination of three parameters, we obtain the best value of every evaluation indicator in three methods.

Method	P	R	F	MRR	MAP
TPR	0.617	0.404	0.465	0.670	0.405
cTPR	0.625	0.406	0.473	0.675	0.407
Our method	<b>0.700</b>	<b>0.440</b>	<b>0.518</b>	<b>0.713</b>	<b>0.440</b>

Table 1. Comparisons of our method and the baselines (t-test, p-value<0.0001)

The comparison of our method to the baselines is shown in table 1. Our method achieves a 7.5% improvement in Precision over the cTPR and 8.3% over the TPR, also increased by more than 3.3% in other indicators.

We also investigate the influence of different parameter values. Due to space limitation, we only provide comparison analysis on MRR. As shown in Fig.3, the bar chart illustrates that the comparisons of our method to the baselines on MRR when  $\lambda$  is set 0.1, 0.3, 0.5, 0.7, 0.9, while

the curve diagram describes the fluctuation of the methods. Although our method is influenced by parameter  $\lambda$ , ours is superior to the baselines in all parameter values.

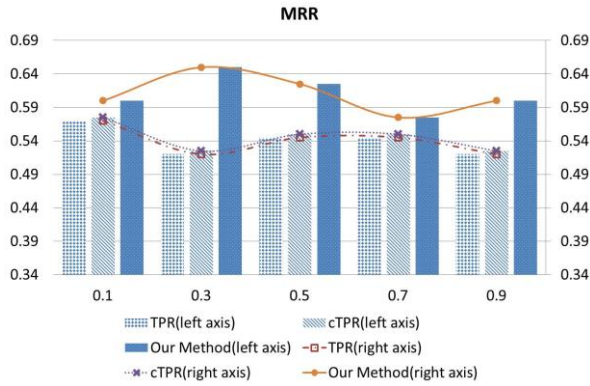


Figure 3. Comparison of the methods on MRR while varying  $\lambda$  (t-test, p-value<0.0001)

Similarly, we can see clearly from Fig.4 that significant promotion of our method compared to the baselines through introducing  $\rho$ . In addition, our method keeps stable with the variations of  $\rho$ .

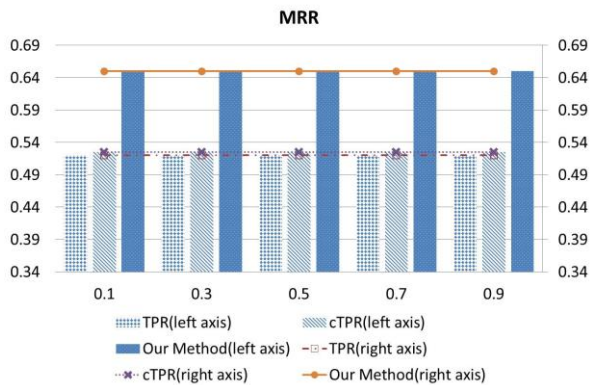


Figure 4. Comparison of the methods on MRR while varying  $\rho$  (t-test, p-value<0.0001)

The statistics in Fig.5 illustrate that our method remains stable when Q is from 0.1 to 0.7, and the curve improves significantly by increasing Q until Q reaches 0.9. While the other two methods change greatly as Q varies. Especially, the baseline methods become rather poor when Q is equal to 0.9. We infer that the baseline methods may lose many links in the tag graph when faced with a high threshold Q, because the link between two tags which the global semantic similarity lower than Q will be removed. Nevertheless, our method can deal with this problem very

well. These experimental results demonstrate the robustness of our method.

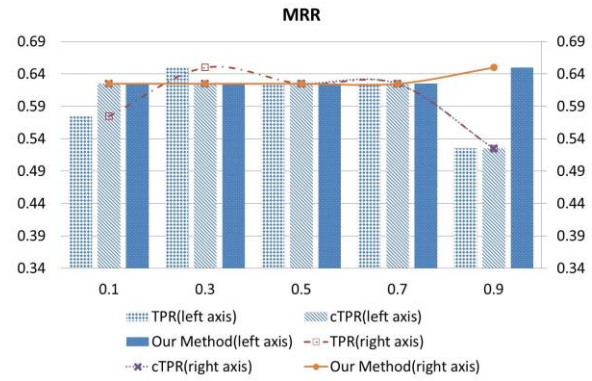


Figure 5. Comparison of the methods on MRR while varying Q (t-test, p-value<0.0001)

#### 4.5 Qualitative Evaluation

In this subsection, some qualitative evaluations are provided on the basis of the resultant graphs with respect to different topics generated by our method.

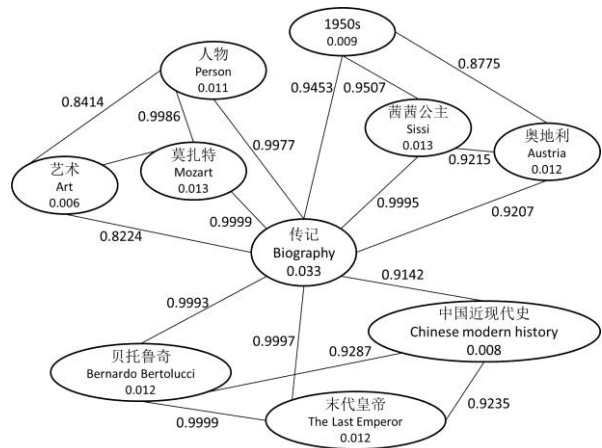


Figure 6. An example of topical graphs generated by our method

The vertex is composed of the tag name and the topical importance score, while the link weight value stands for the degree of semantic similarity between two vertices. As shown in Fig. 6 about biography topic, we can intuitively observe that the vertex '传记' 'Biography' stands in the center while surrounded by a wealth of connectivity. Compared with other vertices, it has the highest importance score in this topic and on behalf of this topic. These observations also confirm the effectiveness of our method.

To our surprise, it seems that some apparently unrelated tags are closely connected in our resultant graph. The interesting findings help us to further detect the fact that usually obtained through common sense or reasoning. For example, 末代皇帝‘The Last Emperor’ and 贝纳尔多·贝托鲁奇‘Bernardo Bertolucci’, which can be used to infer the fact that ‘The Last Emperor’ is a biographical film directed by ‘Bernardo Bertolucci’. Likewise, we observe that the dense connections among 茜茜公主‘Sissi’, 奥地利‘Austria’ and “1950s” can help us to infer the fact that ‘Sissi’ is a “1950s” film in ‘Austria’. All of them are connected to ‘Biography’, which means ‘The Last Emperor’ and ‘Sissi’ all belong to ‘Biography’. These insights further prove that our method can well connect the most related tags together with respect to the topic through the novel link weight model.

In addition, a few unusual genres of movie emerge in our works which enrich the traditional movie categories. For example, 公路‘road movie’, 默片‘silent movie’, 黑色电影‘film noir’, and 神片 shen-pian and so on. The sensibility for upcoming concepts indicates that our method is a necessary complement for traditional concept extraction.

#### 4.6 Error Analysis

We perform error analysis after experiments. A typical error is 姜文‘Jiang Wen’, a famous movie actor in China which is wrongly recognized as a topical key concept. The vertices that closely related to 姜文‘Jiang Wen’ in the graph are 宁静‘Jing Ning’, 夏雨‘Yu Xia’ and 阳光灿烂的日子‘In the Heat of the Sun’. We believe that the best topical key concept about this topic is 文艺‘literature’. However, ‘literature’ cannot be created if it never appears in the tags of Douban.com. This error is due to randomness of folksonomy tagging itself. We consider integrating other relative folksonomy data sources such as Baidu video<sup>3</sup> to overcome this defect in the future work.

## 5 Conclusion

In this paper we study the novel problem of topical key concept extraction from folksonomy. A new link weight function is proposed to improve graph-based ranking method for topical key concept extraction. Quantitative and qualitative

evaluations indicate the robustness and effectiveness of our method. In the future, we will make full use of the topical key concepts and relevant entities, and also the relationships by-product for Chinese ontology construction. Experiments on the folksonomy data from Douban.com Movie show that our method is feasible. We will further explore our method in other domains such as music and more large-scale data with the help of other folksonomy-based systems.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 61273321, 61133012 and the National 863 Leading Technology Research Project via grant 2012AA011102. Finally, we would like to thank the anonymous reviewers for their insightful comments.

## References

- Angeletou S, Sabou M, Motta E. 2008. Semantically Enriching Folksonomies with FLOR. In *the 1st International Workshop on Collective Semantics: Collective Intelligence & the Semantic Web*, Tenerife, Spain, pages 1-16.
- Blei D M, Ng A Y, Jordan M I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, volume 3, pages 993-1022.
- Buitelaar P, Cimiano P, Magnini B. 2005. *Ontology Learning from Text: Methods, Applications and Evaluation*, volume 123, pages 3-12.
- Cui G, Lu Q, Li W, et al. 2009. Automatic acquisition of attributes for ontology construction. In *Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*, Springer Berlin Heidelberg, pages 248-259.
- Haveliwala T H. Topic-sensitive pagerank. 2002. In *Proceedings of the 11th Association for Computing Machinery international conference on World Wide Web (ACM)*, pages 517-526.
- Hearst M A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Association for Computational Linguistics conference on Computational linguistics-Volume 2 (ACL)*, pages 539-545.
- Hotho A, Jäschke R, Schmitz C, et al. 2006. Information retrieval in folksonomies: Search and rank-

<sup>3</sup> <http://video.baidu.com>

- ing. *The Semantic Web: Research and Applications*, Springer Berlin Heidelberg, pages 411-426.
- Hulth A. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Association for Computational Linguistics conference on Empirical methods in natural language processing (ACL)*, pages 216-223.
- Jin Y, Li R, Wen K, et al. 2011. Topic-based ranking in Folksonomy via probabilistic model. *Journal of Artificial Intelligence Review*, 36(2), pages 139-151.
- Liu X, Song Y, Liu S, et al. 2012. Automatic taxonomy construction from keywords. In *Proceedings of the 18th Association for Computing Machinery international conference on Knowledge discovery and data mining (ACM SIGKDD)*, pages 1433-1441.
- Liu Z, Huang W, Zheng Y, et al. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the Association for Computational Linguistics Conference on Empirical Methods in Natural Language Processing (ACL)*, pages 366-376.
- Marlow C, Naaman M, Boyd D, et al. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings of the 17th Association for Computing Machinery conference on Hypertext and hypermedia (ACM)*, pages 31-40.
- Mihalcea R, Tarau P. 2004. TextRank: Bringing order into texts. In *Proceedings of Association for Computational Linguistics Conference on Empirical Methods in Natural Language Processing*, 4(4), pages 404-411.
- Nakayama K, Pei M, Erdmann M, et al. 2008. Wikipedia Mining-Wikipedia as a Corpus for Knowledge Extraction. In *Proceedings of Annual Wikipedia Conference*, pages 1-15.
- Nie L, Davison B D, Qi X. 2006. Topical link analysis for web search. In *Proceedings of the 29th annual international Association for Computing Machinery conference on Research and development in information retrieval (ACM SIGIR)*, pages 91-98.
- Page L, Brin S, Motwani R, Winograd T. 1999. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford Digital Library Technologies Project*, pages 1-17.
- Trant J. 2009. Studying Social Tagging and Folksonomy: A Review and Framework. *Journal of Digital Information*, 10(1), pages 1-42.
- Voorhees E M. The TREC-8 question answering track report. 1999. In *Proceedings of TREC*, pages 77-82.
- Voorhees E, Harman D, Standards N I, et al. 2005. TREC: Experiment and evaluation in information retrieval. *Cambridge: MIT press Boston*, pages 1-567.
- Zhao X, Jiang J, He J, et al. 2011. Topical keyphrase extraction from Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pages 1-10.
- Zhou M, Bao S, Wu X, et al. 2007. An unsupervised model for exploring hierarchical semantics from social annotations. *Journal of the Semantic Web*, pages 680-693.

# Uncovering Distributional Differences between Synonyms and Antonyms in a Word Space Model

Silke Scheible, Sabine Schulte im Walde and Sylvia Springorum

Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart

Pfaffenwaldring 5b, 70569 Stuttgart, Germany

{scheible,schulte,riestesa}@ims.uni-stuttgart.de

## Abstract

For many NLP applications such as Information Extraction and Sentiment Detection, it is of vital importance to distinguish between synonyms and antonyms. While the general assumption is that distributional models are not suitable for this task, we demonstrate that using suitable features, differences in the contexts of synonymous and antonymous German adjective pairs can be identified with a simple word space model. Experimenting with two context settings (a simple window-based model and a ‘co-disambiguation model’ to approximate adjective sense disambiguation), our best model significantly outperforms the 50% baseline and achieves 70.6% accuracy in a synonym/antonym classification task.

## 1 Introduction

One notorious problem of distributional similarity models is that they tend to not only retrieve words that are strongly alike to each other (such as *synonyms*), but also words that differ in their meaning (i.e. *antonyms*). It has often been argued that this behaviour is due to the distributional similarity of synonyms and antonyms: despite conveying different meanings, antonyms also seem to occur in very similar contexts (Mohammad et al., 2013).

In many applications, such as information retrieval and machine translation, the presence of antonyms can be devastating (Lin et al., 2003). While a number of approaches have addressed the issue of synonym and antonym distinction from a computational point of view, they are usually limited in some way, for example by requiring the antonymous words to co-occur in certain patterns (Lin et al., 2003; Turney, 2008), or by relying on external resources such as thesauri (Mohammad et

al., 2013; Yih et al., 2012). Probably due to this strong similarity of their contexts, there have been no successful attempts so far to distinguish the two relations via a standard distributional model such as the word space model (Sahlgren, 2006).

Prominent work in psycholinguistics, however, has shown that humans are able to distinguish the contexts of antonymous words, and that these are by no means interchangeable (Charles and Miller, 1991). The goal of our research is to show that using suitable features these differences can be identified via a simple word space model, relying on contextual clues that govern the ability to distinguish the relations in context. For this purpose, we present a word space model that exploits window-based features for synonymous and antonymous German adjective pairs. Next to investigating the contributions of the various parts-of-speech with regard to the word space model, we experiment with two context settings: one that takes into account all contexts in which the members of the word pairs occur, and one where we approximate context disambiguation by applying ‘co-disambiguation’: establishing the set of nouns that are modified by both members of the pair, and only including distributional information from contexts in which the adjectives premodify one of the set of shared nouns. Two different scenarios relying on Decision Trees then assess our main hypothesis, that the contexts of adjectival synonyms and antonyms are distinguishable from each other.

Our paper is structured as follows: Section 2 reviews some of the theoretical and psycholinguistic hypotheses and findings concerning synonymy and antonymy, and Section 3 reviews previous approaches to synonym/antonym distinction. Based on these theoretical and practical insights, we introduce our hypotheses and approach in Section 4. Section 5 describes the data and implementation of the word space model used in our experiments, and, finally, in Section 6 we discuss our findings.



## 2 Theoretical background

Synonymy and antonymy are without doubt two of the most well-known semantic relations between words, and can be broadly defined as words that are ‘similar’ in meaning (synonyms), and words that are ‘opposite’ in meaning (antonyms).<sup>1</sup> The fascinating issue about antonymy is that even though antonymous words are said to be opposites, they are nevertheless semantically very similar. Cruse (1986) observes that there is a notion of simultaneous closeness and distance from one another, and notes that this can be partially explained by the fact that opposites share the same semantic dimension. For example, the antonyms *hot* and *cold* share the dimension ‘TEMPERATURE’, but unlike synonyms, which are located at identical or close positions on the dimension (such as *hot* and *scorching*), antonyms occupy opposing poles (cf. the schematic representation in Figure 1). Antonymous words are thus similar in all respects but one, in which they are maximally opposed (Willners, 2001).



Figure 1: Semantic dimension

There has been extensive work on linguistic and cognitive aspects of synonyms and antonyms (Lehrer and Lehrer, 1982; Cruse, 1986; Charles and Miller, 1989; Justeson and Katz, 1991). Both relations have played a special role in the area of distributional semantics, which investigates how the statistical distribution of words in context can be used to model semantic meaning. Many approaches in this area are based on the *distributional hypothesis*, that words with similar distributions have similar meanings (Harris, 1968).

In a seminal study, Rubenstein and Goodenough (1965) provided support for the distributional hypothesis for synonyms by comparing the collocational overlap of sentences generated for 130 target words (i.e. 65 word pairs ranging from highly synonymous to semantically unrelated) with synonymy judgements for the pairs, showing that there is a positive relationship between the degree of synonymy between a word pair and the degree to which their contexts are similar. The

<sup>1</sup>In the following, we work with this simple definition of the two relations. For an account of other, more complex definitions, please refer to Murphy (2003).

situation for antonyms with respect to the distributional hypothesis has however been less clear. In fact, Charles and Miller (1991) used the contextual distribution of antonyms to argue *against* the reliability of the co-occurrence approach: they measured how often antonyms co-occur within the same sentence (for example, in contrastive constructions such as ‘either x or y’), and show that the co-occurrence counts for antonyms such as *big/little*, and *large/small* in the Brown corpus are larger than chance.<sup>2</sup> Charles and Miller claim that the fact that antonyms tend to co-occur in the same contexts constitutes a true counter-example to the co-occurrence approach: they display high contextual similarity, but are of low semantic similarity.

As an alternative to the co-occurrence approach, Charles and Miller (1991) proposed a technique based on *substitutability* (cf. also Deese (1965)). Here, the contextual similarity of synonyms/antonyms is determined by presenting human subjects with sentences in which the occurrences of the two words have been blanked out, and by assessing the amount of confusion between the words when asking the subjects which word belongs in which context. While, as anticipated, the level of confusion was high for synonyms, subjects rarely confused the sentential contexts of antonyms, contrary to Charles and Miller’s expectations. They had assumed that direct antonyms<sup>3</sup> such as *strong/weak*, or *powerful/faint*, were interchangeable in most contexts, based on the insight that any noun phrase that can be modified by one member of the pair can also be modified by the other. However, human subjects were very efficient at identifying the correct antonym.

Charles and Miller’s findings suggest that in contrast to synonyms, whose distributional properties are similar, there are clear contextual differences that allow humans to distinguish between the members of an antonym pair. In this paper we aim to show that these differences can be detected with a simple distributional word space model, thereby refuting the claim that antonyms are a counter-example to the co-occurrence approach.

## 3 Previous computational approaches

Due to their special status as both ‘similar’ and ‘different’, work in computational linguistics has sometimes included antonymy under the heading

<sup>2</sup>Similar results were found by Justeson and Katz (1991).

<sup>3</sup>Commonly associated adjectives (Paradis et al., 2009).

of semantic similarity. Recent research however has called for a strict distinction between *semantic similarity* (where entities are related via likeness) and *semantic relatedness* (where dissimilar entities are related via lexical or functional relationships, or frequent association), cf. Budanitsky and Hirst (2006). Accordingly, antonyms fall into the broader category of ‘semantic relatedness’, and should not be retrieved by measures of semantic similarity. That this is of crucial importance was highlighted by Lin et al. (2003), who noted that in many NLP applications the presence of antonyms in a list of similar words can be devastating.

A variety of measures have been introduced to measure semantic similarity, for example by drawing on lexical hierarchies such as WordNet (Budanitsky and Hirst, 2006). In addition, there are corpus-based measures that attempt to identify semantic similarities between words by computing their distributional similarity (Hindle, 1990; Lin, 1998). While these are efficient at retrieving synonymous words, they fare less well at identifying antonyms as non-similar words, and routinely include them as semantically similar words. However, despite the problems resulting from this, there have only been few approaches that explicitly tackle the problem of synonym/antonym distinction, rather than focussing on only synonyms (e.g. Edmonds (1997)) or antonyms (e.g. de Marneffe et al. (2008)).

Lin et al. (2003), who implemented a similarity measure to retrieve distributionally similar words for constructing a thesaurus, were one of the first to propose methods for excluding retrieved antonyms. Lin’s measure uses dependency triples to extract distributionally similar words. In a post-processing step, they filter out any words that appear with the patterns ‘from X to Y’ or ‘either X or Y’ significantly often, as these patterns usually indicate opposition rather than synonymy. They evaluate their technique on a set of 80 synonym and 80 antonym pairs randomly selected from Webster’s Collegiate Thesaurus that are also among their top-50 list of distributionally similar words, and achieve an F-score of 90.5% in distinguishing between the two relations.

Turney (2008) also tackles the task of distinguishing synonyms from antonyms as part of his approach to identifying analogies. Like Lin et al. (2003), he relies on a pattern-based approach, but instead of hand-coded patterns, his algorithm

uses seed pairs to automatically generate contextual patterns (in which both related words must appear). Using ten-fold cross-validation, his approach achieves an accuracy of 75.0% on a set of 136 ‘synonyms-or-antonyms’ questions, compared to a majority class baseline of 65.4%.

A recent study by Mohammad et al. (2013), whose main focus is on the identification and ranking of opposites, also discusses the task of synonym/antonym distinction. Using Lin (2003) and Turney (2008)’s datasets, they evaluate a thesaurus-based approach,<sup>4</sup> where word pairs that occur in the same thesaurus category are assumed to be close in meaning and marked as synonyms, while word pairs occurring in contrasting thesaurus categories or paragraphs are marked as opposites. To determine contrasting thesaurus categories, Mohammad et al. rely on what they call the ‘contrast hypothesis’. Starting with a set of seed opposites across thesaurus categories, they assume that all word pairs across the respective contrasting categories are also contrasting word pairs. The method achieves 88% F-measure on Lin et al. (2003)’s dataset (compared to Lin’s 90.5%), and 90% F-measure on Turney (2008)’s set of ‘synonyms-or-antonyms’ questions, an improvement of 15% compared to Turney’s results.

While all three approaches perform fairly well, they all have certain limitations. Mohammad et al. (2013)’s approach requires an external structured resource in form of a thesaurus. Both Lin et al. (2003) and Turney (2008)’s methods require antonyms to co-occur in fixed patterns, which may be less successful for lower-frequency antonyms. Incidentally, Lin et al. (2003)’s antonyms were chosen from a list of high-frequency terms to increase the chances of finding them in one of their patterns, while Turney (1998)’s data was drawn from websites for Learner English, and is therefore also likely to consist of higher-frequency words.<sup>5</sup> Our proposed model is not subject to such limitations: it does not require external structured resources or co-occurrences in fixed patterns.

## 4 Approach

**Our hypotheses** So far, there have been no successful attempts to distinguish synonymy and antonymy via standard distributional models such as the word space model (Sahlgren, 2006). This

<sup>4</sup>Yih et al. (2012) is another thesaurus-based approach.

<sup>5</sup>Mohammad et al. (2013) show that Lin et al. (2003)’s patterns have a low coverage for their antonym set.



is likely to be due to the assumed similarity of their contexts: Mohammad et al. (2013), for example, state that measures of distributional similarity typically fail to distinguish synonyms from semantically contrasting word pairs. They back up this claim with their own findings: Applying Lin (1998)'s similarity measure to a set of highly-contrasting antonyms, synonyms, and random pairs they show that both the high-contrast set and the synonyms set have a higher average distributional similarity than the random pairs. Interestingly, they also found that, on average, the set of opposites had a higher distributional similarity than the synonyms.

From an intuitive viewpoint such results are surprising: according to Charles and Miller (1991)'s substitutability experiments, there must be contextual clues that allow humans to distinguish between synonyms and antonyms. It appears, however, that these contextual differences are not captured by current measures of semantic similarity, leading to the assumption that synonyms and antonyms are distributionally similar and the claim that antonyms are counter-examples to the distributional hypothesis (cf. Section 2). The goal of our research is to show that this assumption is incorrect, and that contextual differences can be identified via standard distributional approaches using suitable features. In particular, we aim to provide support for the following hypotheses:

- **Hypothesis A.** The contexts of adjectival synonyms and antonyms are *not* distributionally similar.
- **Hypothesis B.** Not all word classes are useful for modelling the contextual differences between adjectival synonyms and antonyms.

We claim that the assumption that synonyms and antonyms are distributionally similar is incorrect. Their distributions may well be similar with respect to certain features (namely the ones commonly used in similarity measures), but our goal is to show that it is possible to identify distributional features that allow an automatic distinction between synonyms and antonyms (Hypothesis A). In particular, we expect synonyms to have a higher level of distributional similarity than antonyms (contrary to Mohammad et al. (2013)'s findings).

We further hypothesise that only some word classes are useful for modelling the contextual differences between adjectival synonyms and antonyms (Hypothesis B). For this purpose, we

plan to investigate the influence of the following parts of speech in our distributional model: adjectives (ADJ), adverbs (ADV), verbs (VV), and nouns (NN). Our prediction is that the class of nouns will not be a useful indicator for distributional differences. This is motivated by Charles and Miller (1991)'s substitutability experiment, in which they claim that a noun phrase that can incorporate one adjective can also incorporate its antonym. As nouns relate to the semantic dimension denoted by the adjectives (cf. Section 2), they are in fact likely to co-occur with both the synonym (SYN) and the antonym (ANT) of a given target (T), resulting in high mutual information values for both, but not necessarily expressing the potential semantic differences between them:

- T: *unhappy* {**man, woman, child, ...**}
- SYN: *sad* {**man, woman, child, ...**}
- ANT: *happy* {**man, woman, child, ...**}

We expect to find more meaningful distributional differences in the *contexts* of such adjective-noun pairs, as illustrated in a simplified example:

- T: *unhappy man* – {*cry, moan, lament, ...*}
- SYN: *sad man* – {*cry, frown, moan, ...*}
- ANT: *happy man* – {*smile, laugh, sing, ...*}

We would for example assume that the set of verbs co-occurring with the target *unhappy* is more similar to the set of verbs co-occurring with its synonym *sad* than to the sets of verbs co-occurring with the antonym *happy*, resulting in higher similarity values for the pair of synonyms than for the pair of antonyms.

**Addressing polysemy** Addressing polysemy is an important task in distributional semantics, both with regard to type-based and token-based word senses: a distributional vector for a word type comprises features and associated feature strengths across all word senses, and a distributional vector for a word token does not indicate a sense if no disambiguation is performed. In recent years there have been a number of proposals that explicitly address the representation and identification of multiple senses in vector models, such as (Erk, 2009; Erk and Padó, 2010; Reisinger and Mooney, 2010; Boleda et al., 2012), with some focussing on identifying predominant word senses, such as (McCarthy et al., 2007; Mohammad and Hirst, 2006). In our experiments, we also aim to incorporate methods for dealing with multiple word senses.

In the task of synonym/antonym distinction, polysemy plays a central role as semantic relations tend to hold between specific *senses* of words rather than between *word forms* (cf. Mohammad et al. (2013)). For adjectives, polysemy directly relates to the semantic dimension they express. For example, depending on the dimension denoted by *hot* (cf. Section 2) we may expect different synonyms and antonyms. If we position *hot* on the dimension of TEMPERATURE, we might expect *scorching* as a synonym, and *cold* as an antonym. However, when *hot* is used to describe a person, we might instead use *attractive* as synonym, and *unattractive* as antonym. In their experiments on adjective synonym and antonym generation, Murphy and Andrew (1993) found that there was indeed considerable context sensitivity depending on the nouns that were modified by the target adjectives, with different synonyms and antonyms being generated.

Based on these insights we experiment with two different context settings: one that takes into account *all* contexts in which the target word and its synonym/antonym occur ('All-Contexts'), and one where we aim to resolve polysemy by applying the method of 'co-disambiguation' ('Codis-Contexts'). The co-disambiguation method attempts to exclude contexts of unrelated senses from consideration by establishing the set of nouns that are modified by both members of the synonym/antonym pair, and only including distributional information from contexts in which the adjectives co-occur (premodify) one of the set of shared nouns. This approach is motivated by the way in which humans might identify the semantic dimension of a pair of synonyms or antonyms out of context: using one member to disambiguate the other by figuring out which common property they express. For example, we intuitively realise that the synonyms *sweet* and *cute* are not related via the dimension of TASTE (as *sweet* might otherwise imply), but are used to describe a pleasing disposition. The co-disambiguation approach attempts to model this strategy by first identifying the nouns shared by the two adjectives across the corpus (such as *sweet/cute* {kid, dog, cottage, ...}), and then only collecting distributional information from such contexts. In the experiments described in the next sections we investigate if this smaller, but more focussed set of contexts can improve the results of our standard 'All-Contexts' model.

## 5 Experimental setup

This section provides an overview of the experimental setup and the distributional model we implemented to test our hypotheses. We work with German data in these experiments, but expect that the findings extend to other languages.

### 5.1 Training and test data

Our dataset is part of a collection of semantically related word pairs compiled via two separate experiments hosted on Amazon Mechanical Turk (AMT)<sup>6</sup>. The experiments were based on a set of 99 target adjectives which were selected from the lexical database GermaNet<sup>7</sup> using a stratified sampling technique accounting for 16 semantic categories, three polysemy classes, and three frequency classes. The first experiment asked AMT workers to propose synonyms, antonyms, and hypernyms for each of the targets. In the second experiment, workers were asked to rate the resulting pairs for the strength of antonymy, synonymy, and hypernymy between them, on a scale between 1 (minimum) and 6 (maximum). Both experiments resulted in 10 solutions per task.

To validate the generated synonym and antonym pairs, we carried out an assessment of their rating means (calculated over 10 ratings per word pair). The results show that there is a highly negative correlation between them with a Pearson  $r$  value of -0.895. This means that the higher a pair's rating as antonym, the lower its rating as synonym, and vice versa, which corresponds to our intuition that synonymy and antonymy are mutually exclusive relations. Figure 2 illustrates the relationship by plotting the average antonym and synonym ratings of all pairs in the dataset against each other.

For the current study we selected 97 synonym and 97 antonym pairs from this data as follows:

- The pairs have a rating means of  $\geq 5$ , representing strong examples of the respective relation types. This narrowed the set of 99 adjective targets to 91 targets, participating in 116 antonym pairs and 145 synonym pairs.
- To decrease sparse data problems we excluded pairs where at least one of its members had a token frequency of  $< 20$  in the sDeWaC-v3 corpus (Faaß et al., 2010), removing 6 antonym and 4 synonym pairs.

<sup>6</sup><https://www.mturk.com>

<sup>7</sup><http://www.sfs.uni-tuebingen.de/lsd>

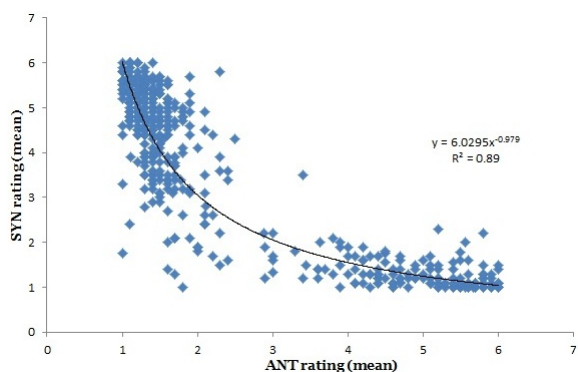


Figure 2: Scatter plot of rating means

- To allow a target-based assessment (cf. Section 5.2), our dataset was reduced to those targets which participate in at least one synonymy and one antonymy relation: 63 targets in total; examples are shown in Table 1. Note that the synonym and antonym pairs of a given target are not necessarily located on the same semantic dimension, as illustrated by the target *süß* (‘sweet’).
- Based on these targets, we sampled an equal number of synonym and antonym pairs from the set, including at least one synonym and one antonym relation for each target, and giving preference to pairs with higher rating means. The resulting set includes 97 synonym and 97 antonym pairs altogether.

Target	Synonym	Antonym
<i>fett</i> (‘fat’)	<i>dick</i> (‘thick’)	<i>dünn</i> (‘thin’)
<i>süß</i> (‘sweet’)	<i>niedlich</i> (‘cute’)	<i>sauer</i> (‘sour’)
<i>dunkel</i> (‘dark’)	<i>düster</i> (‘gloomy’)	<i>hell</i> (‘light’)

Table 1: Dataset examples

## 5.2 Distributional model

**Overview** The main goal of this research is to show that there are distributional differences between synonym and antonym pairs that allow an automatic distinction between them (cf. Hypothesis A). The automatic method we use to address this task is an implementation of the word space model (Sahlgren, 2006; Turney and Pantel, 2010; Erk, 2012) where the members of the word pairs are represented as vectors in space, using contextual co-occurrence counts as vector dimension elements. The distributional similarity of two words is then calculated by means of the cosine function (a standard way of measuring vector similarity in word space models), which quantifies similarity

by measuring the angle between two vectors  $v_T$  and  $v_{SYN}$  (or  $v_{ANT}$ ) in vector space:

$$sim_{COS}(v_T, v_{SYN}) = \frac{v_T \cdot v_{SYN}}{|v_T| \cdot |v_{SYN}|}$$

Following from the discussion in Section 4, we expect higher cosine similarity values for synonyms, and lower values for antonyms. We establish the effectiveness of our proposed model for synonym/antonym distinction by means of an automatic classifier on the set of relation pairs introduced in Section 5.1.

**Co-occurrence information** The co-occurrence information included in the model is drawn from the sDeWaC-v3 corpus (Faaß et al., 2010), a cleaned version of the German web corpus deWaC<sup>8</sup>, which contains around 880 million tokens and has been parsed with Bohnet’s MATE dependency parser (Bohnet, 2010). The corpus further provides lemma and part-of-speech annotations (STTS tagset). We varied the window sizes we took into account as co-occurrence information; here we report our findings for the best window size of 5 tokens to the left and right of the adjectives (but not crossing sentence boundaries).

Instead of simple co-occurrence frequencies, our model uses *local mutual information (LMI)* scores as vector values. LMI is a measure from information theory that compares observed frequencies  $O$  with expected frequencies  $E$ , taking marginal frequencies into account:  $LMI = O \times \log \frac{O}{E}$ , with  $E$  representing the product of the marginal frequencies over the sample size.<sup>9</sup> In comparison to (pointwise) mutual information (Church and Hanks, 1990), LMI improves the well-known problem of propagating low-frequent events through multiplying mutual information by the observed frequency.

**Experimental settings** To address our hypothesis that only some word classes are useful for modelling the contextual differences between adjectival synonyms and antonyms (Hypothesis B), we build separate word spaces for the following collocate types: adjectives (ADJ), adverbs (ADV), verbs (VV), and nouns (NN). In addition, we also consider a combination of all four word classes (COMB). For this purpose, we compiled co-occurrence vectors for each word class by counting the frequencies of all adjective–collocate tuples that appeared in the sDeWaC corpus within

<sup>8</sup><http://wacky.sslmit.unibo.it/doku.php?id=corpora>

<sup>9</sup>See <http://www.collocations.de/AM/> for a detailed illustration of association measures (incl. LMI).

the specified window (here, size 5). For example, the model ‘VV in window w5’ includes all verbs that appear in a context window of five words from the adjectives, such as [*süß* – *verspeisen* – 3 – 12.4448] (‘sweet’ – ‘devour’ – frequency – LMI).

As discussed in Section 4, we consider two context settings: one that collects co-occurrence information from *all* contexts of the adjectives (‘All-Contexts’), and one that applies co-disambiguation to address polysemy (‘Codis-Contexts’). For the latter, word vectors only include co-occurrence information from contexts in which the members of a synonym/antonym pair modify a shared noun.

**Classifier** To establish whether there are significant distributional differences between synonyms and antonyms, and to assess the discriminative power of the different word class models, we experimented with several WEKA<sup>10</sup> classifiers and measures (e.g. Jaccard) and assessed their performance at synonym/antonym distinction using 10-fold cross-validation. Here we describe the results of the best-performing combination of classifier and measure: a Decision Tree classifier (‘J48’) with one single feature (*standard-cosine*, or *cosine-difference* values). Thus, for each of the experimental settings described above we run the classifier twice. In the first scenario, we use the plain cosine values (i.e. the distributional similarity values of the synonym/antonym pairs) as features in the classification. This default scenario is somewhat unrealistic, as it assumes a specific cosine cut-off value that distinguishes synonyms from antonyms. The second scenario addresses this issue and refers to a target-based point of view: It may be the case that for the majority of targets, the cosine values of their synonyms are significantly higher than those of their antonyms, indicating clear distributional differences. However, such information is lost when training the classifier on *all* cosine values in cases where the cosine value of the antonym of a target  $T_1$  is greater than the synonym value of another target  $T_2$ , as illustrated in Figure 3, making it difficult to find an appropriate cut-off value to split the data in classes. We take this into consideration as follows: for each synonym and antonym pair involving target T (cf. Section 5.1), we calculate the difference between their cosine values and use these difference values as input to the classifier. For exam-

ple, the cosine values for the synonym pair *süß* – *niedlich* and antonym pair *süß* – *sauer* (cf. Table 1) are 0.94 (T:SYN) and 0.18 (T:ANT), respectively, and the difference value is calculated as (T:SYN – T:ANT). The resulting value (which may be positive or negative) is used as input for the synonym pair (here, 0.76), while the negated value is used as input for the antonym pair (-0.76). For cases where several synonym or antonym pairs are available, an average difference value is calculated.

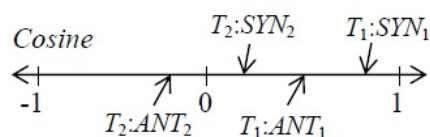


Figure 3: Relative cosine values

## 6 Results

This section presents the results of the Decision Tree classification of synonyms vs. antonyms, using *standard-cosine* values as features (Figure 4) and using *cosine-difference* values (Figure 5). The graphs show the performance of the classifiers in % accuracy for the five part-of-speech-based word space models (ADJ, ADV, NN, VV, and COMB), while at the same time comparing the performances of the two context settings ‘Codis-Contexts’ (dark bars) and ‘All-Contexts’ (light bars). The results are compared against a 50% baseline (dotted line), and significant improvements are marked with a star.

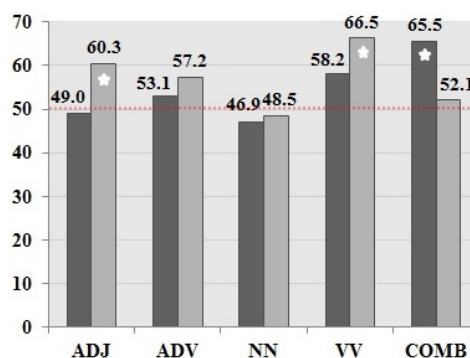


Figure 4: Classification results (*standard-cosine*)

## 7 Discussion

**Hypothesis A** The graphs in Figures 4 and 5 clearly show that it is possible to automatically distinguish between synonymy and antonymy by means of a word space model, with significant improvements over the 50% baseline. These results support our hypothesis that synonyms and antonyms are *not* distributionally similar, and refute the claim that antonyms constitute a counterexample to the distributional hypothesis. An in-

<sup>10</sup>[www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka)

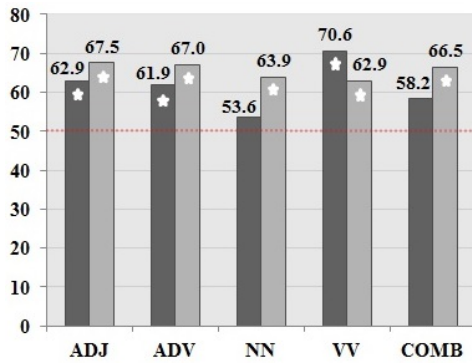


Figure 5: Classification results (*cosine-difference*)

investigation of the decision trees underlying the best-performing classifiers in Figure 4 further shows surprisingly clearly that there *is* a cut-off point over the cosine values that separates synonyms from antonyms, with antonyms in the lower-value and synonyms in the higher-value partition. For example, the cut-off value for the ‘All-Contexts’ model for verbs (light bar in Figure 4) is 0.1186, and any instances with lower cosine values are labelled as antonyms, and with higher values as synonyms, achieving 66.5% accuracy. This is in line with our prediction that synonyms are more distributionally similar than antonyms.

**Hypothesis B** Our second hypothesis, that not all word classes are useful for modelling the contextual differences between adjectival synonyms and antonyms, is also supported by the findings: the word space models built on the class of collocate verbs (VV) appear to be the best discriminators of the relations overall, outperforming the baseline in all four scenarios shown in Figures 4 and 5. All except one of these improvements are statistically significant.<sup>11</sup> The second-best class according to our statistical analysis is the class of adjectives (ADJ), which outperforms the baseline in three of four scenarios (all three being statistically significant). The class of adverbs occupies middle ground, significantly outperforming the baseline only in the *cosine-difference* scenario. As predicted, the noun class (NN) fares worst in the experiments, only (significantly) beating the baseline in one scenario (*cosine-difference*, ‘All-Contexts’).

**Polysemy** The graphs in Figures 4 and 5 show that in most experiment conditions the ‘All-Contexts’ setting (which incorporates

<sup>11</sup> *standard-cosine*, ‘All-Contexts’:  $\chi^2 = 10.85, p < .001$ ; *cosine-difference*, ‘All-Contexts’:  $\chi^2 = 6.55, p < .05$ ; *cosine-difference*, ‘Codis-Contexts’:  $\chi^2 = 8.18, p < .005$ .

co-occurrence information from all contexts) achieves better results than the ‘Codis-Contexts’ setting (which aims to address polysemy by means of ‘co-disambiguation’). However, in the *cosine-difference* scenario, which aims to provide a more accurate representation of distributional differences, the ‘Codis-Contexts’ setting provides a much clearer picture of the differences between the word classes than the ‘All-Contexts’ setting (with accuracy values ranging from 53.6% for nouns to 70.6% for verbs for the former, and 62.9% for verbs to 67.5% for adjectives for the latter). Furthermore, the overall best result (i.e. relying on verbs in the *cosine-difference* scenario) is achieved in the ‘Codis-Contexts’ setting.

A closer analysis of the vector sizes shows that the performance of the ‘co-disambiguation’ approach might be affected by sparse data. Given a larger source of co-occurrence data, the approach may achieve better results than shown in Figures 4 and 5. Overall, our findings suggest that the ‘co-disambiguation’ approach to dealing with polysemy represents a worthwhile avenue for future research, especially on consideration of its other advantages such as ease of implementation and reduced space requirements.

## 8 Conclusion

Our experiments demonstrated that synonyms and antonyms can be distinguished by means of a distributional word space model, refuting the general assumption that synonyms and antonyms are distributionally similar. With 66.5% and 70.6% accuracy in two different classification settings, our model achieves significant improvements over a 50% baseline, and compares favourably to previous approaches by Turney (2008), who achieved an improvement of 9.6% over his baseline, and Lin et al. (2003), whose method is assumed to only work for high-frequency antonyms.

What are the implications of our findings for distributional semantics? First of all, we have shown that the distributional hypothesis holds true even for antonyms. Secondly, our finding that not all word classes are equally useful for modelling the contextual differences between synonyms and antonyms suggests that the performance of distributional measures may be improved by excluding certain word classes from consideration, depending on the task. Finally, we introduced a simple ‘co-disambiguation’ approach to dealing with polysemy in distributional word space models.



## References

- Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of COLING*, Beijing, China.
- Gemma Boleda, Sabine Schulte im Walde, and Toni Badia. 2012. Modelling Regular Polysemy: A Study on the Semantic Classification of Catalan Adjectives. *Computational Linguistics*, 38(3):575–616.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.
- Walter Charles and George Miller. 1989. Contexts of Antonymous Adjectives. *Applied Psycholinguistics*, 10:357–375.
- Walter Charles and George Miller. 1991. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, 6(1):1–28.
- Alan Cruse. 1986. *Lexical Semantics*. CUP, Cambridge, UK.
- Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding Contradictions in Text. In *Proceedings of ACL-HLT*, pages 1039–1047, Columbus, OH.
- James Deese. 1965. *The Structure of Associations in Language and Thought*. The John Hopkins Press, Baltimore, MD.
- Philip Edmonds. 1997. Choosing the Word most typical in Context using a Lexical Co-occurrence Network. In *Proceedings of ACL*, pages 507–509, Madrid, Spain.
- Katrin Erk and Sebastian Padó. 2010. Exemplar-based Models for Word Meaning in Context. In *Proceedings of ACL*, Uppsala, Sweden.
- Katrin Erk. 2009. Representing Words in Regions in Vector Space. In *Proceedings of CoNLL*, pages 57–65, Boulder, Colorado.
- Katrin Erk. 2012. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653.
- Gertrud Faaß, Ulrich Heid, and Helmut Schmid. 2010. Design and Application of a Gold Standard for Morphological Analysis: SMOR in Validation. In *Proceedings of LREC*, pages 803–810, Valletta, Malta.
- Zellig Harris. 1968. Distributional Structure. In *The Philosophy of Linguistics*, pages 26–47. OUP.
- Donald Hindle. 1990. Noun Classification from Predicate-Argument Structures. In *Proceedings of ACL*, pages 268–275.
- John S. Justeson and Slava M. Katz. 1991. Co-Occurrence of Antonymous Adjectives and their Contexts. *Computational Linguistics*, 17:1–19.
- Adrienne Lehrer and Keith Lehrer. 1982. Antonymy. *Linguistics and Philosophy*, 5:483–501.
- Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying Synonyms among Distributionally Similar Words. In *Proceedings of the IJ-CAI*, pages 1492–1493, Acapulco, Mexico.
- Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING*, Montreal, Canada.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 33(4):553–590.
- Saif Mohammad and Graeme Hirst. 2006. Determining Word Sense Dominance Using a Thesaurus. In *Proceedings of EACL*, Trento, Italy.
- Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing Lexical Contrast. *Computational Linguistics*, 39(3).
- Gregory L. Murphy and Jane M. Andrew. 1993. The Conceptual Basis of Antonymy and Synonymy in Adjectives. *Memory and Language*, 32(3):1–19.
- M. Lynne Murphy. 2003. *Semantic Relations and the Lexicon*. Cambridge University Press.
- Carita Paradis, Caroline Willners, and Steven Jones. 2009. Good and Bad Opposites: Using Textual and Experimental Techniques to Measure Antonym Canonicity. *The Mental Lexicon*, 4(3):380–429.
- Joseph Reisinger and Raymond J. Mooney. 2010. Multi-Prototype Vector-Space Models of Word Meaning. In *Proceedings NAACL*, pages 109–117.
- Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.
- Magnus Sahlgren. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.
- Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Artificial Intelligence Research*, 37:141–188.
- Peter D. Turney. 2008. A Uniform Approach to Analogies, Synonyms, Antonyms, and Associations. In *Proceedings COLING*, pages 905–912, Manchester, UK.
- Caroline Willners. 2001. Antonyms in Context. In *Travaux de Institut de Linguistique de Lund 40*, Lund, Sweden.
- Wen-Tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity Inducing Latent Semantic Analysis. In *Proceedings of the EMNLP and CoNLL*, pages 1212–1222, Jeju Island, Korea.

# Multilingual Word Sense Disambiguation Using Wikipedia

**Bharath Dandala**

Dept. of Computer Science  
University of North Texas  
Denton, TX

BharathDandala@my.unt.edu

**Rada Mihalcea**

Dept. of Computer Science  
University of North Texas  
Denton, TX

rada@cs.unt.edu

**Razvan Bunescu**

School of EECS  
Ohio University  
Athens, OH

bunescu@ohio.edu

## Abstract

We present three approaches to word sense disambiguation that use Wikipedia as a source of sense annotations. Starting from a basic monolingual approach, we develop two multilingual systems: one that uses a machine translation system to create multilingual features, and one where multilingual features are extracted primarily through the interlingual links available in Wikipedia. Experiments on four languages confirm that the Wikipedia sense annotations are reliable and can be used to construct accurate monolingual sense classifiers. The experiments also show that the multilingual systems obtain on average a substantial relative error reduction when compared to the monolingual systems.

## 1 Introduction and Motivation

Ambiguity is inherent to human language. In particular, word sense ambiguity is prevalent in all natural languages, with a large number of the words in any given language carrying more than one meaning. For instance, the English noun *plant* can mean *green plant* or *factory*; similarly the French word *feuille* can mean *leaf* or *paper*. The correct sense of an ambiguous word can be selected based on the context where it occurs, and correspondingly the problem of *word sense disambiguation* is defined as the task of automatically assigning the most appropriate meaning to a polysemous word within a given context.

Two well studied categories of approaches to word sense disambiguation (WSD) are represented by knowledge-based (Lesk, 1986; Galley and McKeown, 2003; Navigli and Velardi, 2005) and data-driven (Yarowsky, 1995; Ng and Lee, 1996; Pedersen, 2001) methods. Knowledge-based methods rely on information drawn from

wide-coverage lexical resources such as WordNet (Miller, 1995). Their performance has been generally constrained by the limited amount of lexical and semantic information present in these resources.

Among the various data-driven WSD methods proposed to date, supervised systems have been observed to lead to highest performance in the Senseval evaluations<sup>1</sup>. In these systems, the sense disambiguation problem is formulated as a supervised learning task, where each sense-tagged occurrence of a particular word is transformed into a feature vector which is then used in an automatic learning process. Despite their high performance, the supervised systems have an important drawback: their applicability is limited to those few words for which sense tagged data is available, and their accuracy is strongly connected to the amount of available labeled data.

In this paper, we address the sense-tagged data bottleneck problem by using Wikipedia as a source of sense annotations. Starting with the hyperlinks available in Wikipedia, we first generate sense annotated corpora that can be used for training accurate and robust monolingual sense classifiers (WIKIMONONSENSE, in Section 2). Next, the sense tagged corpus extracted for the *reference* language is translated into a number of *supporting* languages. The word alignments between the reference sentences and the supporting translations computed by Google Translate are used to generate complementary features in our first approach to multilingual WSD (WIKITRANSSENSE, in Section 3). The reliance on machine translation (MT) is significantly reduced during the training phase of our second approach to multilingual WSD, in which sense tagged corpora in the supporting languages are created through the interlingual links available in Wikipedia. Separate classifiers are

<sup>1</sup><http://www.senseval.org>

trained for the reference and the supporting languages and their probabilistic outputs are integrated at test time into a joint disambiguation decision for the reference language (WIKIMUSENSE, in Section 4).

Experimental results on four languages demonstrate that the Wikipedia annotations are reliable, as the accuracy of the WIKIMONONSENSE systems trained on the Wikipedia dataset exceeds by a large margin the accuracy of an informed baseline that selects the most frequent word sense by default. We also show that the multilingual sense classifiers WIKITRANSSENSE and WIKIMUSENSE significantly outperform the WIKIMONONSENSE systems (Section 5).

## 2 The WikiMonoSense System

In an effort to alleviate the sense-tagged data bottleneck problem that affects supervised learning approaches to WSD, the WIKIMONONSENSE system uses Wikipedia both as a repository of word senses and as a rich source of sense annotations. Wikipedia is a free online encyclopedia, representing the outcome of a continuous collaborative effort of a large number of volunteer contributors. Virtually any Internet user can create or edit a Wikipedia webpage, and this “freedom of contribution” has a positive impact on both the quantity (fast-growing number of articles) and the quality (potential mistakes are quickly corrected within the collaborative environment) of this online resource. Wikipedia editions are available for more than 280 languages, with a number of entries varying from a few pages to three millions articles or more per language.

A large number of the concepts mentioned in Wikipedia are explicitly linked to their corresponding article through the use of links or piped links. Interestingly, these links can be regarded as *sense annotations* for the corresponding concepts, which is a property particularly valuable for words that are ambiguous. In fact, it is precisely this observation that we rely on in order to generate sense tagged corpora starting with the Wikipedia annotations (Mihalcea, 2007; Dandala et al., 2012).

### 2.1 A Monolingual Dataset through Wikipedia Links

Ambiguous words such as e.g. *plant*, *bar*, or *argument* are linked in Wikipedia to different articles, depending on their meaning in the context

where they occur. Note that the links are *manually* created by the Wikipedia users, which means that they are most of the time accurate and referencing the correct article. The following represent four example sentences for the ambiguous word *bar*, with their corresponding Wikipedia annotations (links):

1. In 1834, Sumner was admitted to the [[bar (law)|bar]] at the age of twenty-three, and entered private practice in Boston.
2. It is danced in 3/4 time (like most waltzes), with the couple turning approx. 180 degrees every [[bar (music)|bar]].
3. Jenga is a popular beer in the [[bar (establishment)|bar]]s of Thailand.
4. This is a disturbance on the water surface of a river or estuary, often cause by the presence of a [[bar (landform)|bar]] or dune on the riverbed.

To derive sense annotations for a given ambiguous word, we use the links extracted for all the hyperlinked Wikipedia occurrences of the given word, and map these annotations to word senses, as described in (Dandala et al., 2012). For instance, for the *bar* example above, we extract five possible annotations: *bar (establishment)*, *bar (landform)*, *bar (law)*, and *bar (music)*.

In our experiments, the WSD dataset was built for a subset of the ambiguous words used during the SENSEVAL-2, SENSEVAL-3 evaluations and a subset of ambiguous words in four languages: English, Spanish, Italian and German. Since the Wikipedia annotations are focused on nouns (associated with the entities typically defined by Wikipedia), the sense annotations we generate and the WSD experiments are also focused on nouns. We also avoided those words that have only one Wikipedia label. This resulted in a set of 105 words in four different languages: 30 for English, 25 for Italian, 25 for Spanish, and 25 for German. Table 1 provides relevant statistics for the corresponding monlingual dataset.

### 2.2 The WikiMonoSense Learning Framework

Provided a set of sense-annotated examples for a given ambiguous word, the task of a supervised WSD system is to automatically learn a disambiguation model that can predict the correct sense



Language	#words	#senses	#examples
English	30	5.3	632
German	25	4.5	550
Italian	25	5.4	815
Spanish	25	4.6	484

Table 1: #words = number of ambiguous words, #senses = average number of senses, #examples = average number of examples.

for a new, previously unseen occurrence of the word. Assuming that such a system can be reliably constructed, the implications are two-fold. First, accurate disambiguation models suggest that the data is reliable and consists of correct sense annotations. Second, and perhaps more importantly, the ability to correctly predict the sense of a word can have important implications for applications that require such information, including machine translation and automatic reasoning.

The WIKIMONOSENSE system integrates local and topical features within a machine learning framework, similar to several of the top-performing supervised WSD systems participating in the SENSEVAL-2 and SENSEVAL-3 evaluations. The disambiguation algorithm starts with a preprocessing step, where the text is tokenized, stemmed and annotated with part-of-speech tags. Collocations are identified using a sliding window approach, where a collocation is defined as a sequence of words that forms a compound concept defined in Wikipedia. Next, local and topical features are extracted from the context of the ambiguous word. Specifically, we use the current word and its part-of-speech, a local context of three words to the left and right of the ambiguous word, the parts-of-speech of the surrounding words, the verb and noun before and after the ambiguous words, and a global context implemented through sense-specific keywords determined as a list of words occurring at least three times in the contexts defining a certain word sense. We used TreeTagger for part-of-speech tagging<sup>2</sup> and Snowball stemmer<sup>3</sup> for stemming as they both have publicly available implementations for multiple languages. The features are integrated in a Naive Bayes classifier, which was selected for its state-of-the-art performance in previous WSD systems.

<sup>2</sup> [www.cis.uni-muenchen.de/~schmid/tools/TreeTagger](http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger)

<sup>3</sup> [snowball.tartarus.org](http://snowball.tartarus.org)

### 3 The WikiTransSense System

Consider the examples centered around the ambiguous noun “chair”, as shown in Figure 1, where English is the reference language and German is a supporting language. The figure shows only 2 out of the 5 possible meanings from the Wikipedia sense inventory. The two examples illustrate two important ways in which the translation can help disambiguation. First, two different senses of the target ambiguous word may be translated into a different word in the supporting language. Therefore, assuming access to word alignments, knowledge of the target word translation can help in disambiguation. Second, features extracted from the translated sentence can be used to enrich the feature space. Even though the target word translation is a strong feature in general, there may be cases where different senses of the target word are translated into the same word in the supporting language. For example, the two senses “bar (unit)” and “bar (establishment)” of the English word “bar” translate to the same German word “bar”. In cases like this, words in the context of the German translation may help in identifying the correct English meaning.

#### 3.1 A Multilingual Dataset through Machine Translation

In order to generate a multilingual representation for the monolingual dataset, we used Google Translate to translate the data from English into several other languages. The use of Google Translate is motivated by the fact that Google’s statistical machine translation system is available for many languages. Furthermore, through the University Research Program, Google Translate also provides the word alignments. Given a target word in an English sentence, we used the word alignments to identify the position of the target word translation in the translated sentence. Each of the four languages is used as a reference language, with the remaining three used as supporting languages. Additionally, French was added as a supporting language in all the multilingual systems, which means that each reference sentence was translated in four supporting languages.

#### 3.2 The WikiTransSense Learning Framework

Similar to the WIKIMONOSENSE approach described in Section 2.2, we extract the same types

An airline seat is a <u>chair</u> on an airliner in which passengers are accommodated for the duration of the journey. Ein Flugzeugsitz ist ein <u>Stuhl</u> auf einem Flugzeug, in dem Passagiere fr die Dauer der Reise untergebracht sind.
For a year after graduation, Stanley served as <u>chair</u> of belles-lettres at Christian College in Hustonville. Seit einem Jahr nach dem Abschluss, diente Stanley als <u>Vorsitzender</u> Belletristik bei Christian College in Hustonville.

Figure 1: English to German translations from Google Translate, with the target words aligned.

Language	WikiTransSense	WikiMuSense
English	75,832	13,151
German	54,984	8,901
Italian	81,468	4,697
Spanish	48,384	6,560

Table 2: Total number of sentence translations per language, in the two multilingual approaches.

of features from the reference sentence, as well as from the translations in each of the supporting languages. Correspondingly, the feature vector will contain a section with the reference language features, followed by a multilingual section containing features extracted from the translations in the supporting languages. The resulting multilingual feature vectors are then used with a Naive Bayes classifier.

## 4 The WikiMuSense System

The number of sentence translations required to train the WIKITRANSSENSE approach is shown in the second column of Table 2. If one were to train a WSD system for all ambiguous nouns, the large number of translations required may be prohibitive. In order to reduce the dependency on the machine translation system, we developed a second multilingual approach to WSD, WIKIMUSENSE, that exploits the interlingual links available in Wikipedia.

### 4.1 A Multilingual Dataset through Interlingual Wikipedia Links

Wikipedia articles on the same topic in different languages are often connected through interlingual links. These are the small navigation links that show up in the “Languages” sidebar in most Wikipedia articles. For example, the English Wikipedia sense “Bar (music)” is connected through an interlingual link to the German Wikipedia sense “Takt (Musik)”. Given a sense inventory for a word in the reference language, we automatically build the sense repository for a

supporting language by following the interlingual links connecting equivalent senses in the two languages. Thus, given the English sense repository for the word “bar”  $EN = \{bar (establishment), bar (landform), bar (law), bar (music)\}$ , the corresponding German sense repository will be  $DE = \{Bar (Lokal), noteank, NIL, Takt (Musik)\}$ <sup>4</sup>. The resulting sense repositories can then be used in conjunction with Wikipedia links to build sense tagged corpora in the supporting languages, using the approach described in Section 2.1. However, this approach poses the following two problems:

1. There may be reference language senses that do not have interlingual links to the supporting language. In the “bar” example above, the English sense *bar (law)* does not have an interlingual link to German.
2. The distribution of examples per sense in the automatically created sense tagged corpus for the supporting language may be different from the corresponding distribution for the reference language. Previous work (Agirre et al., 2000; Agirre and Martinez, 2004) has shown that the WSD performance is sensitive to differences in the two distributions.

We address the first problem using a very simple approach: whenever there is a sense gap, we randomly sample a number of examples for that sense in the reference language and use Google Translate to create examples in the supporting language. The third column in Table 2 shows the total number of sentence translations required by the WIKIMUSENSE system. As expected, due to the use of interlingual links, it is substantially smaller than the number of translations required in the WIKITRANSSENSE system.

To address the second problem, we use the distribution of reference language as the true distribution and calculate the number of examples to

<sup>4</sup>NIL stands for a missing corresponding sense in German.

be considered per sense from the supporting languages using the statistical method proposed in (Agirre and Martinez, 2004).

#### 4.2 The WikiMuSense Learning Framework

Once the datasets in the supporting languages are created using the method above, we train a Naive Bayes classifier for each language (reference or supporting). Note that the classifiers built for the supporting languages will use the same senses/classes as the reference classifier, since the aim of using supporting language data is to disambiguate a word in the reference language. Thus, for the word “bar” in the example above, if English is reference and German is supporting, the Naive Bayes classifier for German will compute probabilities for the four English senses, even though it is trained and tested on German sentences.

For each classifier, the features are extracted using the same approach as in the WIKIMONOSENSE system.

At test time, the reference sentence is translated into all four supporting languages using Google Translate. The five probabilistic outputs – one from the reference ( $P_R$ ) and four from the supporting classifiers ( $P_S$ ) – are combined into an overall disambiguation score using Equation 1 below. Finally, disambiguation is done by selecting the sense that obtains the maximum score.

$$P = P_R + \sum_S P_S * \min(1, |D_S|/|D_R|) \quad (1)$$

In Equation 1,  $D_R$  is the set of training examples in the reference language  $R$ , whereas  $D_S$  is the set of training examples in a source language  $S$ . When the number of training examples in a supporting language is smaller than the number of examples in the reference language, the probabilistic output from the corresponding supporting classifier will have a weight smaller than 1 in the disambiguation score, and thus a smaller influence on the disambiguation output. In general, the influence of the supporting classifier will always be less than or equal with the influence of the reference classifier.

## 5 Experimental Evaluation

We ran 10-fold cross-validation experiments on the Wikipedia dataset <sup>5</sup>, with all three systems: WIKIMONOSENSE (WMS), WIKITRANSSENSE

<sup>5</sup>The dataset is available from <http://lit.csci.unt.edu>.

Language	MFS	WMS	WTS	WMuS
English	62.2	78.9	<b>81.9</b>	81.3
German	69.5	81.2	84.6	<b>85.6</b>
Italian	66.0	81.8	84.0	<b>84.7</b>
Spanish	66.8	76.0	78.7	<b>79.7</b>

Table 3: WSD macro accuracies.

Language	MFS	WMS	WTS	WMuS
English	59.2	79.3	80.6	<b>80.3</b>
German	75.6	83.9	86.5	<b>87.0</b>
Italian	74.3	84.6	86.3	<b>87.5</b>
Spanish	72.6	79.8	81.1	<b>82.7</b>

Table 4: WSD micro accuracies.

(WTS), and WIKIMUSENSE (WMUS). For the WIKIMUSENSE system, since the gaps in the supporting language datasets are addressed using reference language translations, we enforced the constraint that a translation of the test example does not appear in the training data of the supporting language.

We used two different accuracy metrics to report the performance:

1. *macro accuracy*: an accuracy number was calculated separately for each ambiguous word. Macro accuracy was then computed as the average of these accuracy numbers.
2. *micro accuracy*: the system outputs for all ambiguous words were pooled together and the micro accuracy was computed as the percentage of instances that were disambiguated correctly.

Tables 3 and 4 show the micro and macro accuracies for the three systems. The tables also show the accuracy of a simple WSD baseline that selects the Most Frequent Sense (MFS).

Overall, the Wikipedia-based sense annotations were found reliable, leading to accurate sense classifiers for the WIKIMONOSENSE system with an average relative error reduction of 44%, 38%, 44%, and 28% compared to the most frequent sense baseline in terms of macro accuracy. WIKIMONOSENSE performed better for 76 out of the 105 words in the four languages compared to the MFS baseline, which further indicates that Wikipedia data can be useful for creating accurate and robust WSD systems.

Compared to the monolingual WIKIMONOSENSE system, the multilingual WIKITRANSSENSE system obtained an average relative error reduction of 13.7%, thus confirming the utility of using translated contexts. Relative to the MFS baseline, WIKITRANSSENSE performed better on 83 of the 105 words. Finally, WIKIMUSENSE had an even higher average error reduction of 16.5% with respect to WIKIMONOSENSE, demonstrating that the multilingual data available in Wikipedia can successfully replace the machine translation component during training. Relative to the MFS baseline, the multilingual WIKIMUSENSE system performed better on 89 out of the 105 words.

Since WIKIMUSENSE is still using machine translation when interlingual links are missing, we ran an additional experiment in which MT was completely removed during training to demonstrate the advantage of sense-annotated corpora available in supporting language Wikipedias. Thus, for the 105 ambiguous words, we eliminated all senses that required machine translation to fill the sense gaps. After filtering, 52 words from the four languages had 2 or more sense in Wikipedia for which all interlingual links were available. The results averaged over the 52 words are shown in Table 5 and demonstrate that WIKIMUSENSE still outperforms WIKIMONOSENSE substantially.

Accuracy	WikiMonoSense	WikiMuSense
Macro	83.9	<b>87.2</b>
Micro	87.5	<b>89.8</b>

Table 5: WSD performance with no sense gaps.

We have also evaluated the proposed WSD systems in a coarse-grained setting on the same dataset. Two annotators were provided with the automatically extracted sense inventory from Wikipedia along with the corresponding Wikipedia articles and requested to discuss and create clusters of senses for the 105 words in the four languages. The results on this coarse-grained sense inventory are shown in Tables 6 and 7 indicate that our multilingual systems outperform the monolingual system.

## 5.1 Learning Curves

One aspect that is particularly relevant for any supervised system is the learning rate with respect to the amount of available data. To determine the

	MFS	WMS	WTS	WMuS
English	72.9	87.3	88.9	<b>89.9</b>
German	72.8	84.1	87.8	<b>87.9</b>
Italian	71.7	87.6	89.4	<b>90.0</b>
Spanish	73.6	83.2	86.1	<b>86.9</b>

Table 6: Coarse grained macro accuracies.

	MFS	WMS	WTS	WMuS
English	69.7	88.9	89.7	<b>91.0</b>
German	78.5	86.7	<b>89.6</b>	89.3
Italian	78.4	88.7	90.3	<b>90.9</b>
Spanish	79.8	87	88.7	<b>90.0</b>

Table 7: Coarse grained micro accuracies.

learning curve, we measured the disambiguation accuracy under the assumption that only a fraction of the data were available. We ran 10-fold cross-validation experiments using 10%, 20%, ..., 100% of the data, and averaged the results over all the words in the data set. The learning curves for the four languages are plotted in Figure 2. Overall, the curves indicates a continuously growing accuracy with increasingly larger amounts of data. Although the learning pace slows down after a certain number of examples (about 50% of the data currently available), the general trend of the curve seems to indicate that more data is likely to lead to increased accuracy. Given that Wikipedia is growing at a fast pace, the curve suggests that the accuracy of the word sense classifiers built on this data is likely to increase for future versions of Wikipedia.

Another relevant aspect is the dependency between the amount of data available in supporting languages and the performance of the WIKIMUSENSE system. To measure this, we ran 10-fold cross-validation experiments using all the data from the reference language and varying the amount of supporting language data from 10% to 100%, in all supporting languages. The accuracy results were averaged over all the words. Figure 3 shows the learning curves for the 4 languages. When using 0% fraction of supporting data, the results correspond to the monolingual WIKIMONOSENSE system. When using 100% fraction of the supporting data, the results correspond to the final multilingual WIKIMUSENSE system. We can see that WIKIMUSENSE starts to perform better than WIKIMONOSENSE when

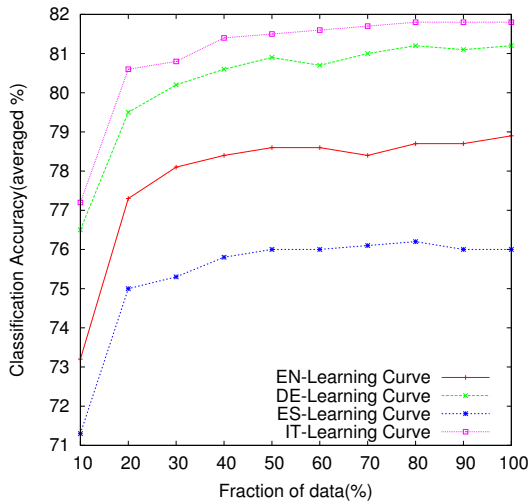


Figure 2: Learning curves for WIKIMONOSENSE.

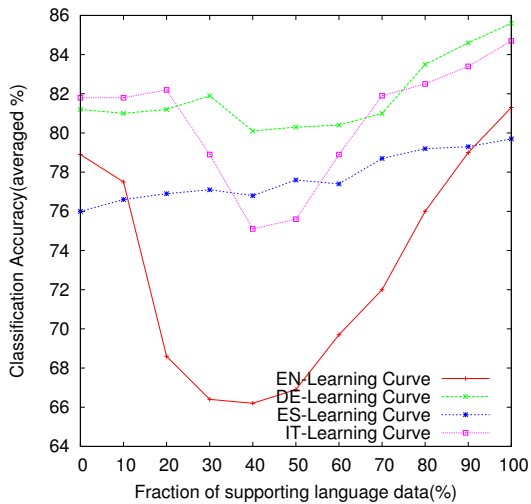


Figure 3: Learning curves for WIKIMUSENSE.

at least 70-80% of the available supporting data is used, and continues to increase its performance with increasing amounts of supporting data.

Finally, we also evaluated the impact that the number of supporting languages has on the performance of the two multilingual WSD systems. Both WIKITRANSSENSE and WIKIMUSENSE are evaluated using all possible combinations of 1, 2, 3, and 4 supporting languages. The resulting macro accuracy numbers are then averaged for each number of supporting languages. Figure 4 indicates that the accuracies continue to improve as more languages are added for both systems.

## 6 Related Work

Despite the large number of WSD methods that have been proposed so far, there are only a few methods that try to explore more than one lan-

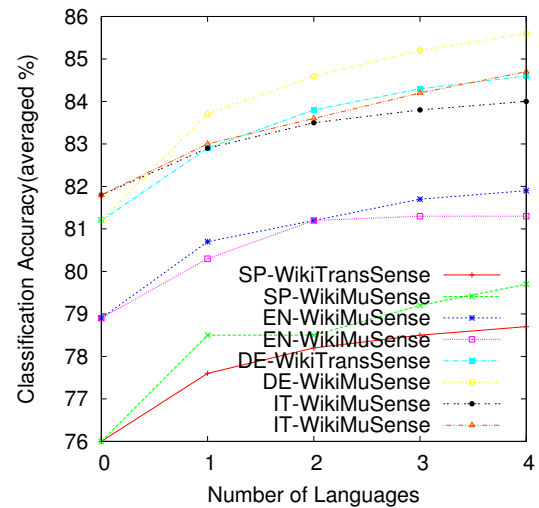


Figure 4: Impact of the number of supporting languages on the two multilingual WSD systems.

guage at a time.

Brown et al. (1991) made the observation that mappings between word-forms and senses may differ across languages and proposed a statistical machine learning technique that exploits these mappings for WSD. Subsequently, several works (Gale et al., 1992; Resnik and Yarowsky, 1999; Diab and Resnik, 2002; Diab, 2004; Ng et al., 2003; Chan and Ng, 2005; Chan et al., 2007) explored the use of parallel translations for WSD.

Li and Li (2004) introduced a bilingual bootstrapping approach, in which starting with in-domain corpora in two different languages, English and Chinese, word translations are automatically disambiguated using information iteratively drawn from the bilingual corpora. Khapra et al. (2009; 2010) proposed another bilingual bootstrapping approach, in which they used an aligned multilingual dictionary and bilingual corpora to show how resource deprived languages can benefit from a resource rich language. They introduced a technique called *parameter projections*, in which parameters learned using both aligned multilingual Wordnet and bilingual corpora are projected from one language to another language to improve on existing WSD methods.

In recent years, the exponential growth of the Web led to an increased interest in multilinguality. Lefever and Hoste (Lefever and Hoste, 2010) introduced a SemEval task on cross-lingual WSD in SemEval-2010 that received 16 submissions. The corresponding dataset contains a collection of sense annotated English sentences for a few words

with their contextually appropriate translations in Dutch, German, Italian, Spanish and French.

Recently, Banea and Mihalcea (2011) explored the utility of features drawn from multiple languages for WSD. In their approach, a multilingual parallel corpus in four languages (English, German, Spanish, and French) is generated using Google Translate. For each example sentence in the training and test set, features are drawn from multiple languages in order to generate more robust and more effective representations known as *multilingual vector-space representations*. Finally, training a multinomial Naive Bayes learner showed that a classifier based on multilingual vector representations obtains an error reduction ranging from 10.58% to 25.96% as compared to the monolingual classifiers. Lefever (2012) proposed a similar strategy for multilingual WSD using a different feature set and machine learning algorithms. Along similar lines, (Fernandez-Ordonez et al., 2012) used the Lesk algorithm for unsupervised WSD applied on definitions translated in four languages, and obtained significant improvements as compared to a monolingual application of the same algorithm. Although these three methodologies are closely related to our WIKITRANSENSE system, our approach exploits a sense inventory and tagged sense data extracted automatically from Wikipedia.

Navigli and Ponzetto (2012) proposed a different approach to multilingual WSD based on *BabelNet* (2010), a large multilingual encyclopedic dictionary built from WordNet and Wikipedia. Their approach exploits the graph structure of *BabelNet* to identify complementary sense evidence from translations in different languages.

## 7 Conclusion

In this paper, we described three approaches for WSD that exploit Wikipedia as a source of sense annotations. We built monolingual sense tagged corpora for four languages, using Wikipedia hyperlinks as sense annotations. Monolingual WSD systems were trained on these corpora and were shown to obtain relative error reductions between 28% and 44% with respect to the most frequent sense baseline, confirming that the Wikipedia sense annotations are reliable and can be used to construct accurate monolingual sense classifiers.

Next, we explored the cumulative impact of features originating from multiple supporting lan-

guages on the WSD performance of the reference language, via two multilingual approaches: WIKITRANSENSE and WIKIMUSENSE. Through the WIKITRANSENSE system, we showed how to effectively use a machine translation system to leverage two relevant multilingual aspects of the semantics of text. First, the various senses of a target word may be translated into different words, which constitute unique, yet highly salient signals that effectively expand the target words feature space. Second, the translated context words themselves embed co-occurrence information that a translation engine gathers from very large parallel corpora. When integrated in the WIKITRANSENSE system, the two types of features led to an average error reduction of 13.7% compared to the monolingual system.

In order to reduce the reliance on the machine translation system during training, we explored the possibility of using the multilingual knowledge available in Wikipedia through its interlingual links. The resulting WIKIMUSENSE system obtained an average relative error reduction of 16.5% compared to the monolingual system, while requiring significantly fewer translations than the alternative WIKITRANSENSE system.

## Acknowledgments

This material is based in part upon work supported by the National Science Foundation IIS awards #1018613 and #1018590 and CAREER award #0747340.

## References

- E. Agirre and D. Martinez. 2004. Unsupervised word sense disambiguation based on automatically retrieved examples: The importance of bias. In *Proceedings of EMNLP 2004*, Barcelona, Spain, July.
- E. Agirre, G. Rigau, L. Padro, and J. Asterias. 2000. Supervised and unsupervised lexical knowledge methods for word sense disambiguation. *Computers and the Humanities*, 34:103–108.
- C. Banea and R. Mihalcea. 2011. Word sense disambiguation with multilingual features. In *International Conference on Semantic Computing*, Oxford, UK.
- P. F. Brown, S. A. Pietra, V. J. Pietra, and R. Mercer. 1991. Word-sense disambiguation using statistical methods. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 264–270. Association for Computational Linguistics.

- Y.S. Chan and H.T. Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3, AAAI'05*, pages 1037–1042.
- Y.S. Chan, H.T. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, Prague, Czech Republic.
- B. Dandala, R. Mihalcea, and R. Bunescu. 2012. Word sense disambiguation using wikipedia. *The People's Web Meets NLP: Collaboratively Constructed Language Resources*.
- M. Diab and P. Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, PA, July.
- M. Diab. 2004. Relieving the data acquisition bottleneck in word sense disambiguation. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, Spain, July.
- E. Fernandez-Ordonez, R. Mihalcea, and S. Hassan. 2012. Unsupervised word sense disambiguation with multilingual representations. In *Proceedings of the Conference on Language Resources and Evaluations (LREC 2012)*, Istanbul, Turkey.
- W. Gale, K. Church, and D. Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the humanities*, 26(5-6):415–439.
- M. Galley and K. McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, Acapulco, Mexico, August.
- M. Khapra, S. Shah, P. Kedia, and P. Bhattacharyya. 2009. Projecting parameters for multilingual word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 459–467.
- M. Khapra, S. Sohoney, A. Kulkarni, and P. Bhattacharyya. 2010. Value for money: balancing annotation effort, lexicon building and accuracy for multilingual wsd. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 555–563.
- E. Lefever and V. Hoste. 2010. Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 15–20. Association for Computational Linguistics.
- E. Lefever. 2012. *ParaSense: parallel corpora for word sense disambiguation*. Ph.D. thesis, Ghent University.
- M.E. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference 1986*, Toronto, June.
- H. Li and C. Li. 2004. Word translation disambiguation using bilingual bootstrapping. *Computational Linguistics*, 30(1):1–22.
- R. Mihalcea. 2007. Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York, April.
- G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11).
- R. Navigli and S. Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.
- R. Navigli and S. P. Ponzetto. 2012. Joining forces pays off: Multilingual joint word sense disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1399–1410, Jeju Island, Korea, July.
- R. Navigli and P. Velardi. 2005. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27.
- H.T. Ng and H.B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL 1996)*, Santa Cruz.
- H.T. Ng, B. Wang, and Y.S. Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan, July.
- T. Pedersen. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2001)*, pages 79–86, Pittsburgh, June.
- P. Resnik and D. Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–134.
- D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995)*, Cambridge, MA, June.

# Semantic v.s. Positions: Utilizing Balanced Proximity in Language Model Smoothing for Information Retrieval\*

Rui Yan<sup>†,‡</sup>, Han Jiang<sup>†,‡</sup>, Mirella Lapata<sup>‡</sup>, Shou-De Lin<sup>\*</sup>, Xueqiang Lv<sup>◇</sup>, and Xiaoming Li<sup>†</sup>

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

<sup>\*</sup>Dept. of Computer Science and Information Engineering, National Taiwan University

<sup>‡</sup>Institute for Language, Cognition and Computation, University of Edinburgh

<sup>◇</sup>Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, BISTU

{r.yan, billybob, lxq, lxm}@pku.edu.cn, mlap@inf.ed.ac.uk, sdlin@csie.ntu.edu.tw

## Abstract

Work on information retrieval has shown that language model smoothing leads to more accurate estimation of document models and hence is crucial for achieving good retrieval performance. Several smoothing methods have been proposed in the literature, using either semantic or positional information. In this paper, we propose a unified proximity-based framework to smooth language models, leveraging semantic and positional information simultaneously in combination. The key idea is to project terms to positions where they originally do not exist (i.e., zero count), which is actually a word count propagation process. We achieve this projection through two proximity-based density functions indicating semantic association and positional adjacency. We balance the effects of *semantic* and *positional* smoothing, and score a document based on the smoothed language model. Experiments on four standard TREC test collections show that our smoothing model is effective for information retrieval and generally performs better than the state of the art.

## 1 Introduction

Recently, statistical language models have attracted much attention in the information retrieval community due to their solid theoretical background as well as their success in a variety of retrieval tasks (Ponte and Croft, 1998; Zhai and Lafferty, 2001b). Queries and documents are assumed to be sampled from hidden generative models, and the similarity between a document and a query is then calculated through the similarity between their underlying language models. Clearly,

good retrieval performance relies on the accurate estimation of the query and document models. As queries are generally too short (Zhai and Lafferty, 2001a), the entire retrieval problem is essentially reduced to the problem of estimating a document language model (Lavrenko and Croft, 2001; Liu and Croft, 2004).

Larger observed data generally allow people to establish a more accurate statistical model. Unfortunately, in retrieval, we often have to estimate a model based on a small sample of data (e.g., a single document or only a few documents). Therefore, given limited data sampling, a language model estimation sometimes encounters with the zero count problem: the maximum likelihood estimator would give unseen terms a zero probability, which is not reliable because a larger sample of the data would likely contain the term. Language model smoothing is proposed to address the problem, and has been demonstrated to affect retrieval performance significantly (Zhai and Lafferty, 2001b).

To this end, the quality of retrieval tasks heavily relies on proper smoothing of the document language model. Although much work on language model smoothing has been investigated, two related retrieval heuristics remain to be further explored: 1) *intra-document* smoothing, a propagation of word count to positions where the term does not exist, within the local document; 2) *inter-document* smoothing, a projection of non-existence terms from the entire collection globally. Both heuristics are implemented in this paper.

As the key idea is to propagate term counts via intra-document and inter-document projection to positions where they originally do not exist, we have two ways of projection: we propose a unified proximity-based framework to smooth language models, formulating semantic and position information simultaneously into a single objective function with balance. Intuitively, a smoothed language model should enhance the coherence between terms with large semantic association, and

This work was partially done when Rui Yan was an intern in Intel Center, National Taiwan University

<sup>†</sup>Indicates equal contributions



analogously for those positional adjacent terms. In other words, the terms that are close to each other (either semantically related or positionally adjacent) should have similar (smoothed) language models; the closer they are, the more similar their smoothed language models are. The smoothing method is based on two density functions of propagated counts of words. Our proposed framework can combine both semantic and positional proximity for intra-/inter-document smoothing naturally, which has not been addressed in the previous works. To the best of our knowledge, we are the first to balance the effect of these proximities for both intra-/inter-document smoothing.

Another main technical challenge lies in how to define the propagation functions of semantic projection and positional adjacency in order to estimate the language model accordingly. As the adjacency function has been carefully explored in (Lv and Zhai, 2009), we mainly focus on proposing and evaluating several different semantic association functions for term propagation. In these density functions, “close-by” terms would receive more propagated counts than “far-away” terms, which captures the proximity heuristics.

We evaluate the retrieval performance using several standard TREC test collections. Experimental results show that our proposed proximity-based smoothing consistently outperforms the baseline smoothing methods, indicating the effectiveness of our approach. The results show that the derived smoothing method can improve over the baseline position-based smoothing method significantly, and either outperform or perform comparably to the corresponding state-of-art semantic proximity-based smoothing method.

The rest of the paper is organized as follows. We start by reviewing previous works. Then we introduce the balanced language model smoothing, based on semantics and positions separately. We describe the experiments and evaluation in the next section and finally draw the conclusions.

## 2 Related Work

Language modeling approaches have recently enjoyed much attention for many different tasks ever since the pioneering work applying on information retrieval (Ponte and Croft, 1998). In the past decade, many variants of language models have been proposed, mostly focusing on improving the estimation of query language models (Zhai and Lafferty, 2001a; Lavrenko and Croft, 2001) and document language models (Liu and Croft, 2004; Tao et al., 2006). These methods

boil down to retrieval functions that implement retrieval heuristics similar to those implemented in a traditional model, such as TF-IDF weighting and document length normalization (Zhai and Lafferty, 2001b). Yet with sound statistical foundation, language models make it easier to optimize parameters and often outperform traditional retrieval models (Song and Croft, 1999).

Due to the importance of smoothing, many approaches have been proposed and tested. To smooth a document language model, most early smoothing methods relied on using a background language model, which is typically estimated based on the whole document collection (Ponte and Croft, 1998; Zhai and Lafferty, 2001b; Miller et al., 1999). In contrast to the simple strategy which smooths all documents with the same background, recently corpus structures have been exploited for more accurate smoothing. The basic idea is to smooth a document language model with the documents similar to the document under consideration through clustering (Liu and Croft, 2004; Xu and Croft, 1999; Mei et al., 2008), document expansion (Kurland and Lee, 2004; Tao et al., 2006), or relevance propagation (Kurland and Lee, 2010; Kurland and Lee, 2006; Qin et al., 2005). All these methods are based on document-level semantics similarity to offer “customized” smoothing for each individual document.

Besides semantics, positional heuristics for retrieval have been examined in (Keen, 1992; Tao and Zhai, 2007; Liu and Croft, 2002; Büttcher et al., 2006). Positional language models are proposed to examine the positional proximity in (Lv and Zhai, 2009; Zhao and Yun, 2009). In their work, the key idea is to define a language model for each position within a document, and score it based on the language models on all its positions: hence the effect of positional adjacency is revealed, while semantic information is hardly incorporated.

There is a study in (Karimzadehgan and Zhai, 2010) which smooths language model by term translation model with backgrounds, while we operate term-to-term association on every term position, which is actually a new granularity. Besides, our method takes both semantic and positional information into account, and formulates the two intrinsically different proximity-based heuristics into a unified term-level smoothing framework. To the best of our knowledge, this is the first approach that achieves the combined smoothing.

### 3 Proximity Based Language Smoothing

We propose a term-level proximity based smoothing approach in the positional language model framework. Each word propagates the evidence of its occurrence to all other positions in the document based on semantic and/or positional projection via density functions. To capture the proximity heuristics, we assign “close-by” words with higher propagated counts than those “far away” from the current word. In other words, most propagated counts come from “nearby” words. Here, *close* and *far* could either be semantic or positional. Each position receives propagated counts of words from an intra-document or an inter-document vocabulary set. All positions have a full vocabulary with different term distributions: each word has a certain non-zero probability to occur in each of the positions, as if all words had appeared in any position with a variety of discounted counts, shown in Figure 1.

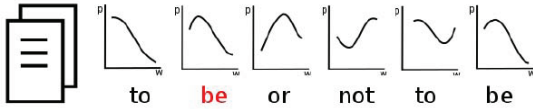


Figure 1: Illustration of different term distributions on different positions for the short document of “To be or not to be”. x-axis denotes all terms in vocabulary, while y-axis indicates the term occurrence probability.

#### 3.1 Semantic Proximity based Propagation

The idea for semantic projection is that if a word  $w$  occurs at position  $i$ , we would like to assume that the highly associated words have also occurred here, with a discounted count. The larger the semantic association is, the larger the propagated count will be. Generally, each propagated count has a value less than 1, which is estimated as the count of  $w$  at position  $i$ .

Let  $d = (w_1, \dots, w_i, \dots, w_j, \dots, w_{|d|})$  be a document, where  $1, i, j$ , and  $|d|$  are absolute positions of the corresponding terms in the document, and obviously  $|d|$  is the length of the document.

$c(w, i, d)$ : the original count of term  $w$  at position  $i$  in document  $d$  before smoothing. If  $w$  occurs at position  $i$ ,  $c(w, i, d)$  is 1, otherwise 0. Similarly,  $c(w, d)$  is the term count in  $d$  and  $c(w)$  is the term count within the collection.

$\phi(w_i, w)$ : the propagated count of  $w$  to position  $i$  based on the existence of  $w_i$ . Intuitively,  $\phi(w_i, w)$  serves as a discounting factor measured by the semantic association between the term  $w_i$  and  $w$ .

$c'(w, i, d)$ : the total propagated count of term  $w$  at position  $i$  from its occurrences in all the positions in the document  $d$ , i.e.,  $c'(w, i, d) = \sum_{j=1}^{|V|} c(w, j, d)\phi(w_i, w) = c(w, d)\phi(w_i, w)$ . Even if  $c(w, i, d)$  is 0,  $c'(w, i, d)$  may be greater than 0.

Note that the **semantic association** function  $\phi(\cdot)$  here is not the same as “similarity”. Generally, association denotes the association between two terms based on the broader background, e.g., co-occurrence or mutual information, etc. Clearly, a major technical challenge for semantic based smoothing lies in a proper model to define the association function. We present here 4 representative association calculations.

**Co-occurrence Likelihood.** Given the term  $w_i$  at position  $i$ , we calculate the co-occurrence probability for the word  $w$  from other positions using:

$$p(w|w_i) = \frac{\#c(w, w_i)}{\#c(w_i)} \quad (1)$$

$\#c(w, w_i)$  is the times of co-occurrence for these two terms. Generally, we need to predefine a sliding window to measure this co-occurrence count, and hence we count  $\#c(w, w_i)$  within the same sentence out of the whole collection.  $\#c(w_i)$  is the term frequency in the document collection.  $p(w|w_i)$  denotes the occurrence probability of  $w$  when  $w_i$  occurs.

Apparently, this definition is asymmetric because  $p(w|w_i) \neq p(w_i|w)$ . When calculate the propagated counts for  $w$ , it is more reasonable to measure the probability given the existence of  $w_i$ . Especially when  $w_i$  is a low-frequency term, we will find the most likely terms with high co-occurrence probability. The semantic association by co-occurrence likelihood is  $\phi_{cl}(w_i, w) = p(w|w_i)$ .

**Mutual Information.** In Information Theory, the mutual information of two random variables is a quantity that measures their mutual dependence, which in our case, is the dependence of co-occurrence probability. The mutual information between the two terms  $w$  and  $w_i$  can be represented as:

$$\text{MI}(w_i, w) = \log \frac{p(w, w_i)}{p(w)p(w_i)} \quad (2)$$

where

$$p(w_i, w) = p(w|w_i)p(w_i) \quad (3)$$

$p(w|w_i)$  is defined in Equation (1), and  $p(w) =$

$\frac{\#c(w)}{\sum_{w' \in V} \#c(w')}$ . Equation (2) can be rewritten as:

$$\text{MI}(w_i, w) = \log \left( \frac{\#c(w, w_i)}{\#c(w) \#c(w_i)} \sum_{w' \in V} \#c(w') \right) \quad (4)$$

Generally, a larger value of mutual information between terms indicates larger association while low value or negative value indicates independency. Although low mutual information is proved to be less dependent, high mutual information does not necessarily guarantee high association, especially for low-frequency terms. Therefore, we apply the Refined Mutual Information (RMI) as an improvement (Manning and Schütze, 1999).

$$\text{RMI}(w_i, w) = \begin{cases} \#c(w, w_i) \text{MI}(w, w_i) \\ 0 \quad (\text{if } \text{MI}(w, w_i) < 0) \end{cases} \quad (5)$$

Finally, we normalize RMI into [0, 1] by using  $\text{RMI}_{\max}$ , the maximum value of RMI, as the semantic association by mutual information:

$$\phi_{mi}(w_i, w) = \frac{\text{RMI}(w, w_i)}{\text{RMI}_{\max}} \quad (6)$$

**Thesaurus-Based Correlation.** A word thesaurus represents the semantic associations of terms, which is often formed into a tree with synonyms, hyponyms and hypernyms modeled by “parent-to-child” relationships, e.g., WordNet<sup>1</sup> or Wikipedia<sup>2</sup>. We illustrate part of WordNet as follows in Figure 2:

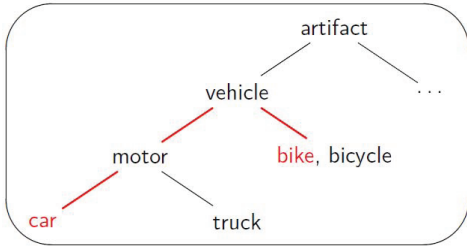


Figure 2: Hierarchical structure of WordNet: red lines imply a possible path between *car* and *bike*.

There could be many paths from one term to the other, and we define the shortest path as the distance between two terms, denoted as  $\text{dist}(w_i, w)$ . Intuitively, the shorter distance is, the larger semantic association is expected. Hence, we utilize a decreasing sigmoid function to model the semantic association based on thesaurus, denoted as  $\phi_{tc}$ :

$$\phi_{tc}(w_i, w) = \frac{1}{1 + e^{\text{dist}(w_i, w)}} \quad (7)$$

<sup>1</sup><http://wordnet.princeton.edu>

<sup>2</sup><http://wikipedia.org>

**Topic Distribution.** “Topics” have long been investigated as the significant latent aspects for linguistic analysis (Hofmann, 2001; Landauer et al., 1998). The utilization of topic models provides a new horizon to investigate the latent correlations between terms and documents. We apply the unsupervised Latent Dirichlet Allocation (Blei et al., 2003) to discover topics<sup>3</sup>. We obtain the probability distribution over topics assigned to a term  $w$ , i.e.,  $p(w|z)$ . The inferred topic representation is the probabilities of terms belonging to the topic  $z$ , which is

$$z = \{p(w_1|z), p(w_2|z), \dots, p(w_i|z)\}$$

We empirically train a  $k$ -topic model ( $k=100$ ) and invert the topic-term representation in Table 1, where each  $w$  is represented as a topic vector  $\vec{w}$ . The semantic association based on topic distribution  $\phi_{td}(w_i, w)$  between  $w_i$  and  $w$  is measured by the cosine similarity on topic vector  $\vec{w}_i$  and  $\vec{w}$ .

$$\phi_{td}(w_i, w) = \frac{\vec{w}_i \cdot \vec{w}}{\|\vec{w}_i\| \|\vec{w}\|} \quad (8)$$

Table 1: Inverted *topic-term* vector representation.

$\vec{w}_1$	$p(w_1 z_1)$	$p(w_1 z_2)$	...	$p(w_1 z_k)$
$\vec{w}_2$	$p(w_2 z_1)$	$p(w_2 z_2)$	...	$p(w_2 z_k)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vec{w}_{ V }$	$p(w_V z_1)$	$p(w_V z_2)$	...	$p(w_V z_k)$

### 3.2 Intra-/Inter-Document Smoothing

For every position  $i$  to estimate the language model, we can project a term from other positions within the document through the defined semantic association functions, namely *intra-document* smoothing. We can also project all terms from the whole vocabulary set to position  $i$  via  $\phi(\cdot)$ , which is actually an *inter-document* smoothing effect from the global collection and hence solve the zero probability problem.

Before smoothing, the original word count distribution for position  $i$  in document  $d$  is  $\mathcal{D}(i, d)$ , with only  $c(w_i, i, d)=1$  while all other items are 0.

$$\mathcal{D}(i, d) = \left[ \underbrace{[c(w_1, i, d), \dots, c(w_i, i, d), \dots, c(w_{|V_d|}, i, d)]}_{V_d}, \underbrace{[0, \dots, 0]}_{V \setminus V_d} \right]$$

After the semantic based intra-document smoothing, the word count distribution becomes:

$$\mathcal{D}_s(i, d) = \left[ \underbrace{[c'(w_1, i, d), \dots, c'(w_i, i, d), \dots, c'(w_{|V_d|}, i, d)]}_{V_d}, \underbrace{[0, \dots, 0]}_{V \setminus V_d} \right]$$

<sup>3</sup>We use Stanford TMT (<http://nlp.stanford.edu/software/tmt/>), with default parameter settings.

Imagine the whole collection as a long *virtual document*, the terms outside the document vocabulary of  $V_d$  could also be smoothed by inter-document smoothing, i.e.,  $c'(w, i) = c(w)\phi(w_i, w)$ . To control the impact of out-of-document vocabulary, we add a parameter  $\mu \in [0, +\infty)$  here:

$$\mathcal{D}_s(i, d) = \left[ \underbrace{[c'(w_1, i, d), \dots, c'(w_{|V_d|}, i, d)]}_{V_d}, \underbrace{[\mu c'(w_j, i), \dots, \mu c'(w_{|V|}, i)]}_{V \setminus V_d} \right]$$

### 3.3 Positional Proximity based Propagation

Analogously, for the positional-based smoothing, the smoothed count by positional proximity is  $c''(w, i, d) = \sum_{j=1}^{|V|} c(w, j, d)\psi(i, j)$ . We apply the best positional proximity based density function of Gaussian projection  $\psi(i, j)$  in (Lv and Zhai, 2009).  $\sigma$  is a fixed parameter here.

$$\psi(i, j) = \exp\left[\frac{-\Delta(i, j)^2}{2\sigma^2}\right] \quad (9)$$

Analogously to the semantic smoothing, we also include the intra-/inter-document smoothing in the positional count propagation. It is natural to measure the distance offset between two terms within the same document. To measure the position distance between terms from different documents, we define  $\Delta(i, j) = +\infty$  when the term  $w$  at a certain position  $j$  is not from the document which contains  $w_i$ , i.e.,

$$\Delta(i, j) = \begin{cases} |i - j| & (w_j \in d) \\ +\infty & (w_j \notin d) \end{cases} \quad (10)$$

In this way, the projection value of  $\psi(i, j)$  is calculated to be 0 when  $w_j \notin d$ . Actually the definition is rather flexible, the value of projection for terms from different documents is easy to adjust to be non-zero when Equation (10) is changed.

The word count distribution  $\mathcal{D}_p(i|d)$  is as follows after positional proximity based smoothing:

$$\mathcal{D}_p(i, d) = \left[ \underbrace{[c''(w_1, i, d), \dots, c''(w_i, i, d), \dots, c''(w_{|V_d|}, i, d)]}_{V_d}, \underbrace{[0, \dots, 0]}_{V \setminus V_d} \right]$$

### 3.4 Balanced Proximity Combination

We can estimate a language model for the position  $i$  based on the propagated counts reaching the position. Since we have two smoothed language distributions, i.e.,  $\mathcal{D}_s(i|d)$  and  $\mathcal{D}_p(i|d)$ , with uniform representation, we can combine both smoothing

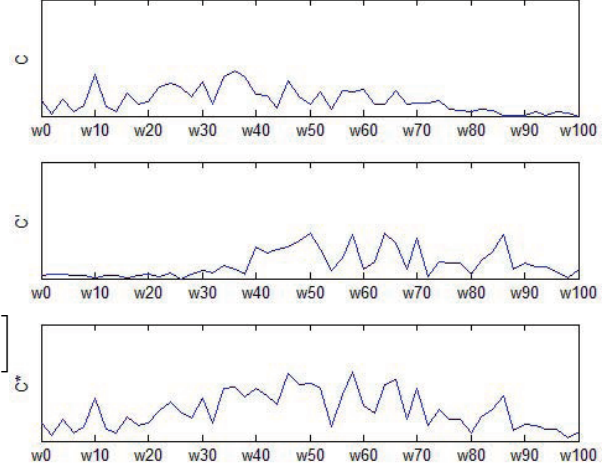


Figure 3: Linear interpolation of two smoothed language models. The upper two word distributions are overlaid into one.

strategies with balance by distribution function superposition, illustrated in Figure 3.

Based on the term propagation, we have a term frequency vector  $\langle w_1, w_2, \dots, w_{|V_d|}, \dots, w_V \rangle$  at position  $i$ , forming a virtual document  $d_i$ .  $\lambda$  is to control the relative contributions from semantic proximity based smoothing and positional proximity based smoothing, formulated as:

$$\begin{aligned} \hat{c}(w, i, d) &= \lambda c'(w, i, d) + (1 - \lambda)c''(w, i, d) \\ &= \lambda \sum_{j=1}^{|V|} c(w, j, d)\phi(w_i, w) + \\ &\quad (1 - \lambda) \sum_{j=1}^{|V|} c(w, j, d)\psi(i, j) \end{aligned} \quad (11)$$

$\phi(w_i, w)$  is to measure the semantic association between  $w$  and the word  $w_i$  from position  $i$ , and  $\psi(i, j)$  is the distance discount factor.

Thus the language model of this virtual document can be estimated as:

$$p(w|i, d) = \frac{\hat{c}(w, i, d)}{\sum_{w' \in V} \hat{c}(w', i, d)} \quad (12)$$

where  $V$  is the vocabulary set.  $\sum_{w' \in V} \hat{c}(w', i, d)$  is actually the length of the virtual document.  $p(w|i, d)$  is the language model at position  $i$ . Thus given a query  $q$ , we can adopt the KL-divergence retrieval model (Lafferty and Zhai, 2001) to score each language model at every position as follows:

$$Score(q, d, i) = - \sum_{w \in V} p(w|q) \log \frac{p(w|q)}{p(w|i, d)} \quad (13)$$

$p(w|q)$  is a query language model. We apply the 1) *best* scoring and 2) *average* scoring of all positions in the document as the retrieval ranking strategy (Lv and Zhai, 2009).

## 4 Experiments

### 4.1 Dataset and Evaluation

In this section, we evaluate the effectiveness of our smoothing strategies empirically. We use four representative TREC data sets: AP (Associated Press news 1988-90), LA (LA Times), WSJ (Wall Street Journal 1987-92) and TREC8 (Disk 4 & 5, the ad hoc data used in TREC8). They represent different sizes and genres, with the same source, queries, and preprocessing procedure as in (Tao et al., 2006; Lv and Zhai, 2009). Table 2 shows the basic statistics of these datasets in detail. We used the title field of a TREC topic description to simulate short keyword queries in our experiments.

Table 2: Detailed basic information of 4 datasets.

	AP	LA	WSJ	TREC8
#doc	242,918	131,896	173,252	528,155
avg(dl)	442.4	492.5	388.7	468.3
qry id	51-150	301-400	51-100 151-200	401-450
#qry	100	100	100	50
#t_qrel	21,819	2,350	10,141	4,728
avg(ql)	4.55	2.63	4.68	2.46

#doc/#qry: number of docs/queries; #t\_qrel: number of relevant docs; avg(dl)/avg(ql): average length of doc/qry.

In each experiment, we first use the baseline model (KL-divergence) to retrieve 2,000 documents for each query, and then use the smoothing methods (or a baseline method) to re-rank them. The top-ranked 1,000 documents for all runs are compared using P@10 and the Mean Average Precisions (MAP) as the main metric.

$$\text{MAP} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{|D_r|} \sum_{i=1}^k P_i \times rel_i$$

$|Q|$  is the number of queries,  $|D_r|$  denotes the total number of relevant documents,  $P_i$  is the precision at  $i$ -th position for, also called P@ $i$  (Manning et al., 2008).  $rel_i$  is an indicator function equaling 1 if the item at rank  $i$  is a relevant document, 0 otherwise.

### 4.2 Algorithms for Comparison

We examine the retrieval performance on the standard datasets. The first baseline group is based on the traditional language model. **LM** is the language model without smoothing at all, while **LM+JM** and **LM+Diri** are to smooth the language model with the whole collection as background information, using Jelinek-Mercer (JM) and Dirichlet (Diri) smoothing methods correspondingly (Zhai and Lafferty, 2001b).

We also examine a series of semantic based language smoothing. The most representative semantic smoothing is the Cluster-Based Document

Model (**CBDM**) proposed by Liu et al. (2004). We apply the default settings for the method, (e.g., clustering methods, etc). Semantic based methods use semantically similar documents as a smoothing corpus for a particular document: CBDM clusters documents and smooths a document with the cluster where that document belongs to. However, this method is only based on document-level semantic similarity rather than term-level semantic association.

We also include Positional Language Model (**PLM**) proposed by Lv et al. (2009), which is the state-of-art positional proximity based language smoothing. PLM mainly utilizes positional information while no semantic association is considered. We implemented the best reported PLM kernel with Dirichlet smoothing from the collection for comparison.

Finally we include our proposed Balanced Proximity-based Model, denoted as **BPM**, which formulates semantic proximity and positional proximity into a unified language smoothing framework, with flexible intra-document smoothing and inter-document smoothing. In all, we have 7 methods to compare their performance.

### 4.3 Overall Performance Comparison

In this section, we compare BPM smoothing with several previously proposed methods, using the Dirichlet smoothing prior which performs best as mentioned in these works. The prior parameter is set at 1000 for all methods to rule out any potential influence of Dirichlet smoothing (Liu and Croft, 2004; Tao et al., 2006). For fairness, we conduct the same pre-processing to all methods. The parameter is chosen by 10-fold cross validation.

For baselines, we use the source code from the original author, and report the results we get. The advantage of CBDM, PLM (and BPM) over the simplest language smoothing with Dirichlet and Jelinek-Mercer smoothing has long been proved. We hence focus on the meaningful comparison between the sophisticated smoothing techniques. Note that under the real scenario, as we could not always predefine which kernel would perform best on a particular dataset, for fairness, we take the average performance of all semantic association kernels as the results of BPM, and the parameters are chosen using 10-fold cross validation described in Section 4.4.2. Tables 3 and 4 show that our model outperforms PLM in MAP and P@10 values on four data sets. The improvement presumably comes from the combination of both semantic and positional proximity based smooth-

MAP	LM	LM+JM	LM+Diri	CBDM	PLM	BPM
AP	0.169133	0.179245	0.180625	0.204361	0.204216	<b>0.207428***</b>
LA	0.204195	0.222077	0.219500	<b>0.240332</b>	0.221190	0.231593
WSJ	0.206986	0.220919	0.221066	0.253834	0.269038	<b>0.277115**</b>
TREC8	0.181040	0.214923	0.209676	0.219018	0.240894	<b>0.248852</b>
P@10	LM	LM+JM	LM+Diri	CBDM	PLM	BPM
AP	0.402020	0.403030	0.403030	<b>0.432323</b>	0.418501	0.400000
LA	0.251020	0.256122	0.245918	0.288776	0.278571	<b>0.289913*</b>
WSJ	0.365142	0.369204	0.379826	0.435036	0.423628	<b>0.446802*</b>
TREC8	0.360508	0.368204	0.358496	<b>0.442242</b>	0.425900	0.438028
Time (in sec)	LM	LM+JM	LM+Diri	CBDM	PLM	BPM
PerQuery	0.151803	0.174667	0.180906	337.08198	0.683829	0.918593

Table 3: Overall performance comparison on MAP and P10 results among all methods. \*, \*\*, \*\*\* indicate that we accept the improvement hypothesis of BPM over the best rival baseline by Wilcoxon test at a significance level of 0.1, 0.05, 0.01 respectively. Efficiency is measured in seconds.

ing; intuitively, the lower bound of BPM is the performance of PLM by tuning the combination parameter  $\lambda$  fixed at 1, which is actually a special case for BPM. It is interesting to find that CBDM based on semantic smoothing performs well in some datasets. We further examine into the datasets of LA and TREC8: in these sets, the semantic proximity weights more than positional proximity, i.e., a smaller  $\lambda$  in Figure 4. As CBDM conducts a more principled way of exploiting semantic smoothing by clustering structures, it should not be too surprising for its performance on datasets which emphasize semantic proximity.

**Efficiency.** The LM group is naturally faster without sophisticated calculations. BPM is a little slower than PLM but with consistent better performance. CBDM shows the lowest efficiency due to mass calculations of similarity for clustering.

#### 4.4 Strategy Analysis

Generally speaking, strategies can be sorted into two categories: component selection and parameter tuning. Each time, we tune one strategy while the other one remains fixed.

##### 4.4.1 Component Selection

There is one substitutive component of designing the semantic propagation function, where the term association can be calculated by co-occurrence likelihood  $\phi_{cl}$ , mutual information  $\phi_{mi}$ , thesaurus-based correlation  $\phi_{tc}$  and topic distribution  $\phi_{td}$ . We examine the performance of different functions to calculate the semantic association and the results are listed in Table 4 and 5.

From the tables above, we can see that most of the semantic association functions have slightly different performance, indicating that these four

MAP	$\phi_{cl}$	$\phi_{mi}$	$\phi_{tc}$	$\phi_{td}$
AP	<b>0.208971</b>	0.207159	0.206713	0.206868
LA	<b>0.231850</b>	0.231557	0.231482	0.231483
WSJ	0.276261	<b>0.278372</b>	0.276829	0.276999
TREC8	0.242348	<b>0.251085</b>	0.250977	0.250996

Table 4: MAP of different semantic associations.

P@10	$\phi_{cl}$	$\phi_{mi}$	$\phi_{tc}$	$\phi_{td}$
AP	<b>0.409091</b>	0.392929	0.398990	0.398991
LA	<b>0.294898</b>	0.285571	0.288592	0.290592
WSJ	<b>0.447102</b>	0.436101	0.446986	0.446019
TREC8	0.438008	<b>0.438103</b>	0.437998	0.438002

Table 5: P@10 of different semantic associations.

measurements are all able to capture the semantic proximity based association among terms. Among all semantic proximity functions, the co-occurrence likelihood  $\phi_{cl}$  performs best in most cases, which means it is reasonable and most natural to smooth the zero count of terms if the co-occurred terms appear.

##### 4.4.2 Parameter Settings

There are two free parameters to tune, i.e.,  $\lambda$  and  $\mu$ .  $\lambda$  is to balance the relative contributions from semantic proximity and positional proximity, while  $\mu$  is to control the weight of inter-document smoothing from the whole collection. Keeping one parameter fixed, we vary the other one to examine the changes of its performance based on all datasets. For each of the 4 datasets, we divide the set and use the 10-fold cross validation to train parameters for testings. We illustrate the performance of parameter sensitivity by tuning  $\lambda$  and  $\mu$  based on all semantic association kernels, as shown in Figure 4.

To control the tradeoff between semantic and positional proximity combination, we gradually



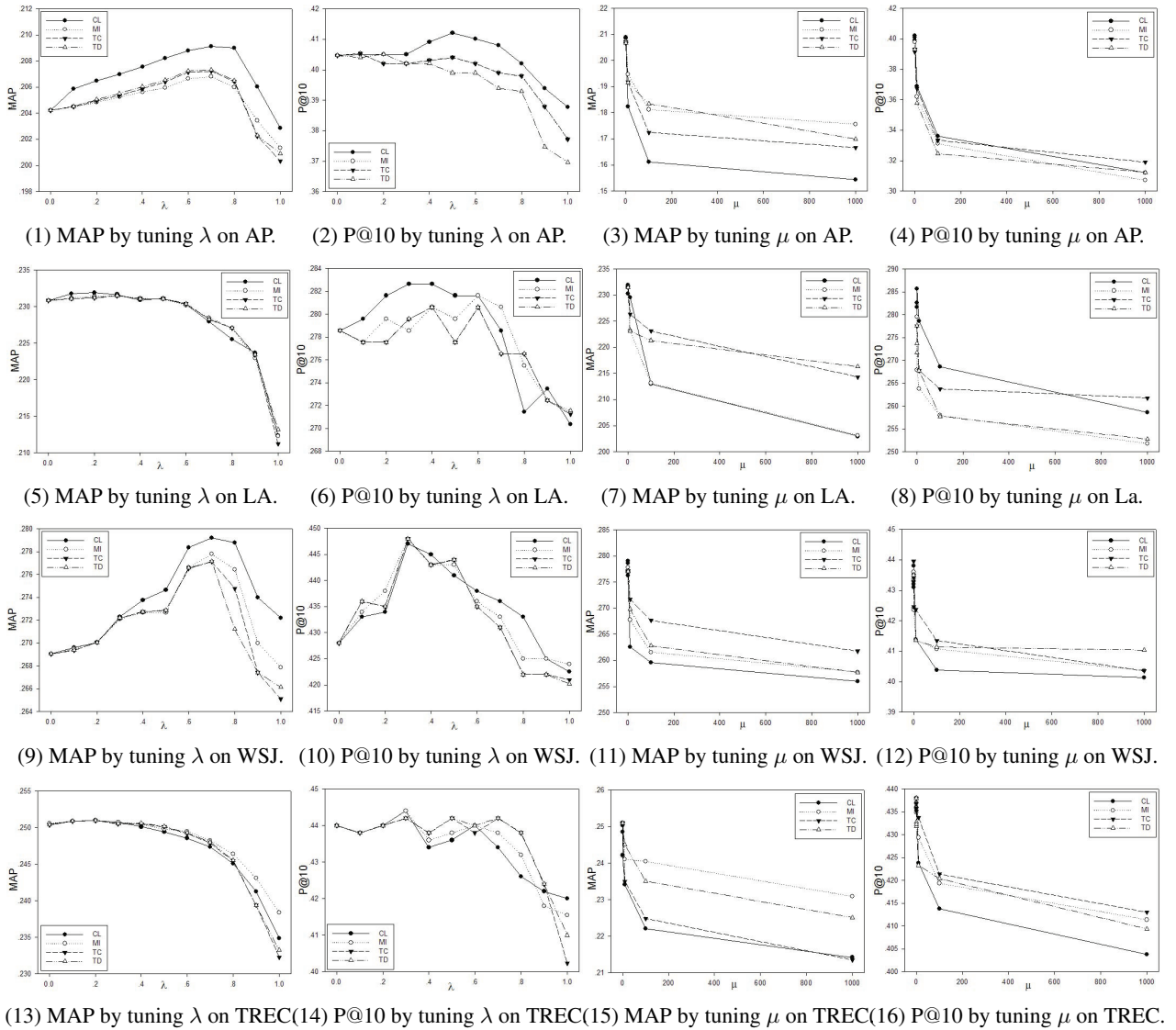


Figure 4: Examine the sensitivity of  $\lambda$  and  $\mu$  by all semantic association functions on all datasets.

change  $\lambda$  from 0 to 1 at the step of 0.1 to examine the effect in Figure 4. The combination of both proximity outperforms the performance in isolation ( $\lambda = 1$  or 0). An interesting observation is that due to the instinct difference of used queries and datasets, the optimal  $\lambda$  varies from one set to another: for AP and WSJ, a larger  $\lambda$  is needed and for LA and TREC8, a small  $\lambda$  is desired perhaps due to the semantic association is more biased for these datasets/corresponding queries: in general, the combination is a better strategy.

We then examine the impact of out-of-document vocabulary controlled by  $\mu$  in Figure 4. Although the performance varies on different datasets as well, for MAP, the performance is generally downward when  $\mu$  grows larger, and for P@10, the performance achieves best when  $\mu$  is relatively small ( $\mu=0.1$  or 0.01), which indicates the impact of inter-document smoothing should not be excessively over introduced.

## 5 Conclusions

In this paper, we combined both semantic and positional proximity heuristics to improve the effect of language model smoothing, which has not been addressed before. We proposed and studied four different semantic proximity-based propagation functions as well as the positional proximity density function to estimate the smoothed language model. Experimental results show that BPM outperforms most alternative baselines in terms of MAP and P@10, which indicates the effectiveness of our proposed method.

Besides the effective fusion of semantic and positional proximity ( $\lambda \neq 0$ ), we further investigate the semantic propagation function, and find that co-occurrence likelihood association performs best. In the future, we will incorporate corpus information such as clustering features into the semantic proximity function for better smoothing.

## Acknowledgments

This work was supported by “III Innovative and Prospective Technologies Project” of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China, and by National Natural Science Foundation of China (Grant No. 61271304) and Key Program of Beijing Municipal Natural Science Foundation (Grant No. KZ201311232037).

## References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.
- Stefan Büttcher, Charles L. A. Clarke, and Brad Lushman. 2006. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of SIGIR '06*, pages 621–622.
- Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196. 10.1023/A:1007617005950.
- Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of SIGIR '10*, pages 323–330.
- E. Michael Keen. 1992. Some aspects of proximity searching in text retrieval systems. *Journal of Information Science*, 18(2):89–98.
- Oren Kurland and Lillian Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of SIGIR '04*, pages 194–201.
- Oren Kurland and Lillian Lee. 2006. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In *Proceedings of SIGIR '06*, pages 83–90.
- Oren Kurland and Lillian Lee. 2010. Pagerank without hyperlinks: Structural reranking using links induced by language models. *ACM Trans. Inf. Syst.*, 28(4):18:1–18:38, November.
- John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR '01*, pages 111–119.
- T.K. Landauer, P.W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.
- Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of SIGIR '01*, pages 120–127.
- Xiaoyong Liu and W. Bruce Croft. 2002. Passage retrieval based on language models. In *Proceedings of CIKM '02*, pages 375–382.
- Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of SIGIR '04*, pages 186–193.
- Yuanhua Lv and ChengXiang Zhai. 2009. Positional language models for information retrieval. In *Proceedings of SIGIR '09*, pages 299–306.
- C.D. Manning and H. Schütze. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.
- C.D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.
- Qiaozhu Mei, Duo Zhang, and ChengXiang Zhai. 2008. A general optimization framework for smoothing language models on graph structures. In *Proceedings of SIGIR '08*, pages 611–618.
- David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of SIGIR '99*, pages 214–221.
- Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of SIGIR '98*, pages 275–281.
- Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, Zheng Chen, and Wei-Ying Ma. 2005. A study of relevance propagation for web search. In *Proceedings of SIGIR '05*, pages 408–415.
- Fei Song and W. Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of CIKM '99*, pages 316–321.
- Tao Tao and ChengXiang Zhai. 2007. An exploration of proximity measures in information retrieval. In *Proceedings of SIGIR '07*, pages 295–302.
- Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings HLT-NAACL '06*, pages 407–414.
- Jinxi Xu and W. Bruce Croft. 1999. Cluster-based language models for distributed retrieval. In *Proceedings of SIGIR '99*, pages 254–261.
- Chengxiang Zhai and John Lafferty. 2001a. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM '01*, pages 403–410.
- Chengxiang Zhai and John Lafferty. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR '01*, pages 334–342.
- Jinglei Zhao and Yeogirl Yun. 2009. A proximity language model for information retrieval. In *Proceedings of SIGIR '09*, pages 291–298.



# An Unsupervised Parameter Estimation Algorithm for a Generative Dependency N-gram Language Model

Chenchen Ding and Mikio Yamamoto

Department of Computer Science

University of Tsukuba

1-1-1 Tennodai, Tsukuba, 305-8573, Japan

{tei@mibel.,myama@}cs.tsukuba.ac.jp

## Abstract

We design a language model based on a generative dependency structure for sentences. The parameter of the model is the probability of a *dependency N-gram*, which is composed of lexical words with four kinds of extra tags used to model the dependency relation and valence. We further propose an unsupervised expectation-maximization algorithm for parameter estimation, in which all possible dependency structures of a sentence are considered. As the algorithm is language-independent, it can be used on a raw corpus from any language, without any part-of-speech annotation, tree-bank or trained parser. We conducted experiments using four languages: English, German, Spanish and Japanese. The results illustrate the applicability and the properties of the proposed approach.

## 1 Introduction

Statistical language models are a fundamental component of speech recognition systems, machine translation systems, and so forth. Presently, the N-gram language model is the most widely used approach. This model focuses on sequences of neighboring lexical words (Fig. 1), and uses the probabilities of these sequences as model parameters. Due to the full lexicalization of the N-gram language model, local features of word sequences can be well modeled. However, an N-gram language model cannot capture relatively long-range features, because it regards a sentence as a flat string and ignores its structure.

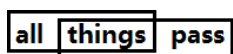


Figure 1: The N-gram language model treats the English sentence “*all things pass*” composed of (*all*, *things*) and (*things*, *pass*), for  $N = 2$ .

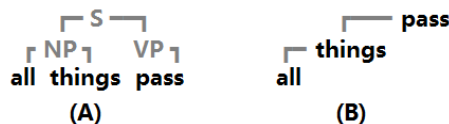


Figure 2: The constituency-based parsing (A) and the dependency-based parsing (B)<sup>1</sup> for the English sentence “*all things pass*”.

On the other hand, revealing the structure of a sentence is the task of parsing, which is based on linguistically oriented formulations and focuses on generating the likeliest structure for a given sentence. For this purpose, there are constituency-based and dependency-based formulations (Fig. 2). The former organizes continuous word sequences in a hierarchy of small range to large range groups with linguistically oriented labels; the latter links words with dependency relations<sup>2</sup>.

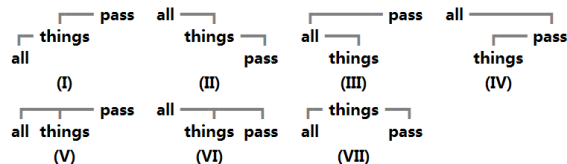


Figure 3: All possible dependency structures for the English sentence “*all things pass*”. (I) is the linguistically correct structure while the original N-gram language model handles the sentence as if it has the structure labeled (II). We consider all these structures in our unsupervised estimation algorithm.

In this paper, we focus on introducing sentence structure into language modeling. We propose a generative dependency N-gram language model that integrates a generative dependency structure of a sentence into the original N-gram language model. We prefer the dependency-based formula-

<sup>1</sup>For the illustrations in this paper, we use the following representation to show the dependency structure. If two aligned words are on different levels, the upper one is the head of the lower one; if they are on the same level, they are siblings.

<sup>2</sup>In general, the dependency relations can be further classified using linguistically oriented labels. However, they are not indispensable and we do not use them in our approach.

tion because it can directly model the relations between words. In the proposed model, the parameter is the probability of the dependency N-gram, which is a sequence of words along the dependency structure rather than along a flat left-to-right string. The proposed model is thus as fully lexical as the original N-gram language model. We further propose an expectation-maximization (EM) algorithm for estimating the probability of arbitrary order<sup>3</sup> dependency N-grams, by considering all possible dependency structures<sup>4</sup> of a sentence (Fig. 3). The proposed algorithm is unsupervised, language-independent and needs no linguistic information.

## 2 Related Work

The technical report by Chen and Goodman (1998) has compared various approaches to the N-gram language model and the modified Kneser-Ney discounting proposed in it is still the state-of-the-art approach. Since the N-gram language model only captures local lexical features, there have been proposals to generalize the lexical N-gram by word-class (Brown et al., 1992) or to model long-range word co-occurrences by word triggers (Tillmann and Ney, 1997). However, these models are unaware of the sentence structure and basically take a sentence as a flat string.

Many approaches have been proposed for constituency-based parsing (Collins, 1998; Klein and Manning, 2003; Klein and Manning, 2004) and for dependency-based parsing (Eisner, 1996; Lee and Choi, 1997; Kudo and Matsumoto, 2002; Klein and Manning, 2004; Nivre, 2008). Presently, discriminative approaches (Kudo and Matsumoto, 2002; Nivre, 2008) are used more than generative ones for dependency-based parsing, because a generative model is usually restricted to being bi-lexical (i.e., the components are bi-grams of head-modifier pairs) and it is hard to handle more lexical information.

There have been some attempts to integrate sentence structure into language modeling. Chelba and Jelinek (2000) have proposed a constituency-based approach, but the use of language-dependent non-terminals cannot be avoided. There are also dependency-based approaches (Stolcke et al., 1997; Gao and Suzuki, 2003; Graham and van Genabith, 2010). However,

these approaches need a trained dependency parser because they construct a language model based on the decisive best structure produced by the parser.

In our approach, we utilize a generative dependency model to guarantee the constituency of a language model<sup>5</sup>, but our model and algorithm can handle arbitrary numbers of lexical words. Furthermore, our approach needs no extra parser to generate the best structure of a sentence but, instead, takes all possible dependency structures into consideration.

## 3 Generative Dependency Model

We model the marginal probability of a sentence  $S$  over the set  $D$  of all possible dependency structures of  $S$ ,  $P(S) = \sum_{d \in D} P(S, d)$ . As described in Klein and Manning (2004), if we separate the dependency structure and lexicalization, then  $\sum_{d \in D} P(S, d) = \sum_{d \in D} P(d)P(S|d)$ . The term  $P(S|d)$  is given by a model of fully lexical word sequences with dependency relations. However, the term  $P(d)$  is difficult to model and is usually taken to be a constant, as in Paskin (2002). To deal with this problem, the *dependency model with valence* (DMV)<sup>6</sup> proposed by Klein and Manning (2004) introduces a special mark *STOP*. However, it is necessary to distinguish two kinds of parameters,  $P_{\text{STOP}}$  and  $P_{\text{CHOOSE}}$  in the bi-gram estimation, which makes it difficult to extend the approach to higher orders.

In a similar approach to that used in the DMV, we introduce four kinds of tags to normalize the distribution of modifier numbers (the valence) of a head word. In this paper, we use  $\langle L \rangle$ ,  $\langle /L \rangle$ ,  $\langle R \rangle$  and  $\langle /R \rangle$  to show the start and end of the left and right modifier word sequences of a head word, respectively. The dependency structure can thus be organized as nested word sequences. Specifically, a modifier word sequences of a head word is in a form of  $M = m_0^{\phi+1} \equiv m_0, m_1, \dots, m_{\phi+1}$ , where  $m_0 \equiv \langle O \rangle$ ,  $m_{\phi+1} \equiv \langle /O \rangle$  ( $O \in \{L, R\}$ ) and  $m_1^{\phi}$  is a lexical  $\phi$ -word sequence. We show an example of the dependency structure in Fig. 4. On the other hand, in contrast to the DMV, we treat the tags as ordinary words in the parameter estimation. So the parameters of our model have a uniform representation, by which our approach can be easily extended to arbitrary high orders.

<sup>3</sup>“Order” here means the number of lexical words ( $N$ ).

<sup>4</sup>Only projective dependency structures are considered.

<sup>5</sup> $\sum_{S \in L} P(S) = 1$  for the set  $L$  composed of all the sentences  $S$  in a language.

<sup>6</sup>A generative model.

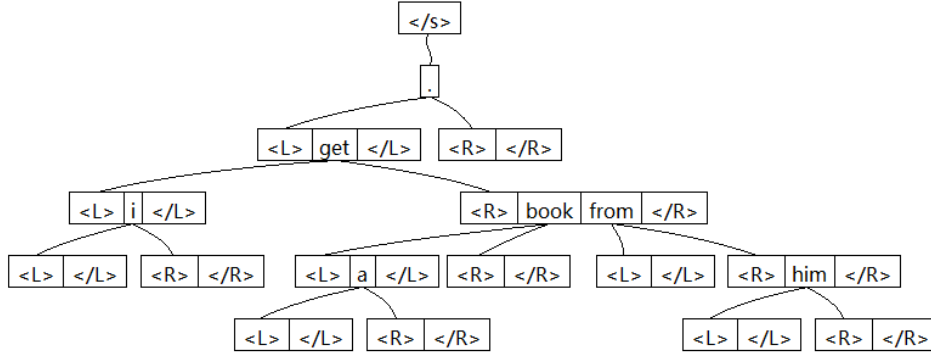


Figure 4: A dependency structure for the English sentence “*i get a book from him .*”, with  $\langle L \rangle$ ,  $\langle R \rangle$ ,  $\langle /L \rangle$ ,  $\langle /R \rangle$  tags. The root of the sentence is marked as  $\langle /s \rangle$  and for a word without modifiers, its modifier word sequences are  $\langle O \rangle \langle /O \rangle$  where  $O \in \{L, R\}$ .

Because our model is essentially equivalent to the generative *Model C* in Eisner (1996), the consistency of the language model can be guaranteed. That is,  $\langle O \rangle m_1^\phi \langle /O \rangle$  ( $O \in \{L, R\}$ ) is generated as a Markov sequence to serve as the modifier word sequences (left/right separately) of the head word. The “start tag”  $\langle O \rangle$  always satisfies  $P(m_0 = \langle O \rangle) \equiv 1$  to represent the nested structure. The “end tag”  $\langle /O \rangle$  terminates the generation process, so: the larger  $P(m_{\phi+1} = \langle /O \rangle)$  is, the smaller  $\phi$ , which is the number of generated words, becomes, and vice versa.

Without loss of generality, the probability of  $m_{\kappa+1}$  ( $0 \leq \kappa \leq \phi$ ) in  $M = m_0^{\phi+1}$  can be represented by  $P(m_{\kappa+1} | m_0^\kappa, H)$ , where  $H$  is the history of  $M$  along the generated path<sup>7</sup>. We use the independent assumption that the probability of a word in the generation process depends on only its direct ancestors and the orientation between them. So, the general probability can be simplified to:

$$P(h^0 | o^1, h^1, \dots, o^{n-1}, h^{n-1}) \quad (1)$$

where  $h^k$  is a lexical word,  $h^{k+1}$  is the head word of  $h^k$ , and  $o^k \in \{\langle L \rangle, \langle R \rangle\}$  is retained in the history to show the dependency orientation.  $\langle /L \rangle$  and  $\langle /R \rangle$  tags can and only can<sup>8</sup> take the place of  $h^0$ .

The sequence  $(h^0, o^1, h^1, \dots, o^{n-1}, h^{n-1})$  in Exp. (1) is referred as a dependency N-gram in this paper. For example, a dependency N-gram is  $(\langle /L \rangle, \langle L \rangle, \text{him}, \langle R \rangle, \text{from}, \langle R \rangle, \text{get}, \langle L \rangle, ., \langle /s \rangle)$  in the dependency structure illustrated in Fig. 4. Exp. (1) is the probability of the dependency N-gram and thus the parameter of our model, where the dependency relation and valence are modeled uniformly for arbitrary order parameters.

<sup>7</sup>The generation process can be realized in a depth-first or a breadth-first way but the distinction is unessential.

<sup>8</sup>Because they cannot have further modifiers.

## 4 Parameter Estimation

### 4.1 Notation

For a sentence  $S = w_0^{l+1} \equiv w_0, w_1, \dots, w_{l+1}$ , where  $w_0 \equiv \langle s \rangle$  and  $w_{l+1} \equiv \langle /s \rangle$ , a dependency N-gram  $(h^0, o^1, h^1, \dots, o^{n-1}, h^{n-1})$  can be denoted by  $\mathbf{d} = (d_0, d_1, \dots, d_{n-1})$  where  $h^k = w_{d_k}$ . That is, a dependency N-gram can be denoted by an N-tuple of the absolute positions of words in a given sentence. As the magnitudes of  $d_k$  and  $d_{k+1}$  show the orientation,  $o^{k+1}$  can be omitted.<sup>9</sup>

Lee and Choi (1997) propose the *complete-link set* and *complete-sequence set* for head-modifier pair (i.e., a dependency bi-gram in our model) to handle all possible projective dependency structures of a sentence in a recursive manner. We follow the terms they use and extend their definitions to adapt them to our dependency N-gram model. We use  $Link(\mathbf{d})$  to denote the complete-link set of an N-tuple  $\mathbf{d}$ , and  $Seq(\mathbf{d})$  for the complete-sequence set.

In Lee and Choi (1997), the complete-link set of a span  $[i, j]$  in a sentence is composed of all possible dependency structures within the span, with the directional dependency link of the two words  $w_i$  and  $w_j$ . The complete-sequence set of a span  $[i, j]$  is defined as the set of all possible sequences with any number (including zero) of adjacent complete-link sets having the same direction within the span. By our notation, i.e. the word at  $d_1$  is the direct head of the word at  $d_0$  for  $Link(d_0, d_1)$ ; but the word at  $d_1$  is an ancestor (not only a direct head) of the word at  $d_0$  for  $Seq(d_0, d_1)$ . The two kinds of sets can be defined recursively and the set of all possible dependency

<sup>9</sup>If  $h^0$  is  $\langle /L \rangle$  or  $\langle /R \rangle$ , we retain them in  $\mathbf{d}$ . The orientations can also be unambiguously omitted for these two tags.

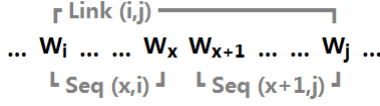


Figure 5:  $Link(\mathbf{d} = (i, j))$ . In Lee and Choi (1997), for a span  $[i, j]$ ,  $Link(i, j)$  is composed of the dependency link of  $w_i$  and  $w_j$ , and all possible pairs of complete-sequence sets  $Seq(x, i)$  and  $Seq(x + 1, j)$ .

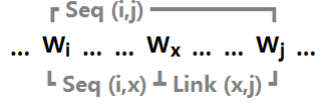


Figure 6:  $Seq(\mathbf{d} = (i, j))$ . In Lee and Choi (1997), for a span  $[i, j]$ ,  $Seq(i, j)$  is composed of all possible pairs of complete-sequence set  $Seq(i, x)$  and complete-link set  $Link(x, j)$ .

structures of a sentence  $S = w_0^{l+1}$  is the complete-sequence set over the span  $[1, l + 1]$ . We illustrate these recursive relations in Fig. 5 and Fig. 6.

Because more than two words are involved in the proposed dependency N-gram, we generalize the two kinds of sets for the N-tuples  $\mathbf{d}$  rather than just spans. The generalization still retains the properties of  $d_0$  and  $d_1$  in  $Link(\mathbf{d})$  and  $Seq(\mathbf{d})$ , as well as the recursive properties of the two kinds of sets. We show examples of a dependency tri-gram in Fig. 7 and Fig. 8.

#### 4.2 Recursive Definition

Here, we give the formulation of the recursive definition of the complete-link set and complete-sequence set for an arbitrary order dependency N-gram. First, due to the properties of the projective dependency structure, any  $d_k$  ( $k \in [1, n - 1]$ ) in the N-tuple  $\mathbf{d} = (d_0, d_1, \dots, d_{n-1})$  needs to satisfy the following constraint to guarantee that a head word is outside of the range covered by a chain of its descendants.

$$\begin{aligned} d_k &> \max(d_0, \dots, d_{k-1}), \text{ or} \\ d_k &< \min(d_0, \dots, d_{k-1}) \end{aligned} \quad (2)$$

Trivially, we take  $\langle /s \rangle$  as the *root* mark of a sentence  $S = w_0^{l+1}$ , and the  $\langle s \rangle$  as the head of itself or the  $\langle /s \rangle$ . So, we have the following constraints.

$$d_{k+1} = 0, \text{ if } d_k = l + 1, \text{ or } d_k = 0 \quad (3)$$

For convenience, we introduce three kinds of operations, *Push*, *Cover*, and *Insert* over an in-

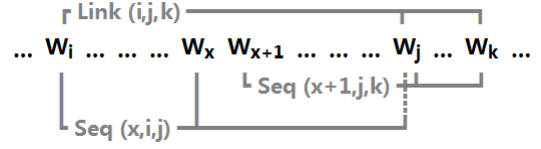


Figure 7:  $Link(\mathbf{d} = (i, j, k))$ . In our model, an extended high-order (3 is shown here) complete-link set  $Link(i, j, k)$  is composed of the N-tuple  $\mathbf{d}$ , and all possible pairs of complete-sequence sets  $Seq(x, i, j)$  and  $Seq(x + 1, j, k)$ .

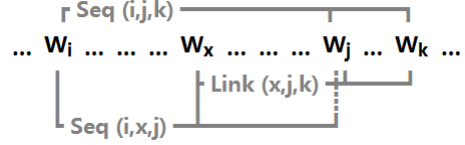


Figure 8:  $Seq(\mathbf{d} = (i, j, k))$ . In our model, an extended high-order (3 is shown here) complete-sequence set  $Seq(i, j, k)$  is composed of all possible pairs of complete-sequence set  $Seq(i, x, j)$  and complete-link set  $Link(x, j, k)$ .

dex  $x$  (absolute word position) and an N-tuple  $\mathbf{d} = (d_0, d_1, \dots, d_{n-1})$ :

$$Push(x, \mathbf{d}) = (x, d_0, d_1, \dots, d_{n-2}) \quad (4)$$

$$Cover(x, \mathbf{d}) = (x, d_1, d_2, \dots, d_{n-1}) \quad (5)$$

$$Insert(x, \mathbf{d}) = (d_0, x, d_2, \dots, d_{n-1}) \quad (6)$$

Then, the  $Link(\mathbf{d})$  and  $Seq(\mathbf{d})$  can be defined by Exp. (7) and Exp. (9) below, where “ $\times$ ” indicates the direct product of sets.

$$\begin{aligned} Link(\mathbf{d}) = & \bigcup_{\substack{\text{if } d_1=l+1, \text{ then } i=d_1-1; \\ \text{else } i \in [\min(d_0, d_1), \max(d_0, d_1)-1]}} \\ & \{Seq(Left(i, \mathbf{d})) \times \\ & Seq(Right(i + 1, \mathbf{d})) \times \mathbf{d}\} \end{aligned} \quad (7)$$

where

$$\begin{aligned} (Left, Right) = & \\ & \begin{cases} (Push, Cover), \text{ if } d_0 < d_1 \\ (Cover, Push), \text{ if } d_0 > d_1 \end{cases} \end{aligned} \quad (8)$$

$$\begin{aligned} Seq(\mathbf{d}) = & \bigcup_{\substack{i \in [\min(d_0, d_1), \max(d_0, d_1)] \\ \text{and } i \neq d_1}} \\ & \{Seq(Insert(i, \mathbf{d})) \times \\ & Link(Cover(i, \mathbf{d}))\} \end{aligned} \quad (9)$$

Exp. (7) shows that a complete-link set is recursively composed of the direct product of all possible complete-sequence set pairs, with the N-tuple  $\mathbf{d}$  itself.<sup>10</sup> Exp. (9) shows that a complete-sequence set is recursively composed of the direct product of all possible pairs of a complete-link set and a smaller complete-sequence set.

To start the recursive definition, we replace  $d_0$  by  $\langle /L \rangle$  and  $\langle /R \rangle$  for all  $Seq(\mathbf{d})$  with  $d_0 = d_1$ <sup>11</sup>.

$$\begin{aligned} Left(x, \mathbf{d}) &= Left(\langle /R \rangle, \mathbf{d}), \\ &\text{if } x = \min(d_0, d_1) \text{ in Exp. (7)} \end{aligned} \quad (10)$$

$$\begin{aligned} Right(x, \mathbf{d}) &= Right(\langle /L \rangle, \mathbf{d}), \\ &\text{if } x = \max(d_0, d_1) \text{ in Exp. (7)} \end{aligned} \quad (11)$$

$$\begin{aligned} Insert(x, \mathbf{d}) &= Push(\langle /L \rangle, \mathbf{d}), \\ &\text{if } x = d_0, \text{ and } d_0 < d_1 \text{ in Exp. (9)} \end{aligned} \quad (12)$$

$$\begin{aligned} Insert(x, \mathbf{d}) &= Push(\langle /R \rangle, \mathbf{d}), \\ &\text{if } x = d_0, \text{ and } d_0 > d_1 \text{ in Exp. (9)} \end{aligned} \quad (13)$$

### 4.3 Estimation

According to the recursive definition, it is natural to derive an inside-outside algorithm (Lari and Young, 1990), which is an adaption of the EM algorithm (Dempster et al., 1977) to tree structures, to conduct parameter re-estimation by calculating the inside and outside probabilities of all complete sets in sentences.

We generalize the expressions in Exp. (7) and Exp. (9) to Exp. (14) and Exp. (15) respectively, where the notation  $\langle \cdot, \cdot \rangle$  stands for an unordered 2-tuple of a complete-set pair.

$$Link(\mathbf{d}) = \bigcup_{\forall \langle Sub_1, Sub_2 \rangle} \{Sub_1 \times Sub_2 \times \mathbf{d}\} \quad (14)$$

$$Seq(\mathbf{d}) = \bigcup_{\forall \langle Sub_1, Sub_2 \rangle} \{Sub_1 \times Sub_2\} \quad (15)$$

We further define  $R_{Link}(Link(\mathbf{d}), \langle Sub_1, Sub_2 \rangle)$  as a relation for  $Link(\mathbf{d}), \langle Sub_1, Sub_2 \rangle$  satisfying Exp. (14). Similarly,  $R_{Seq}(Seq(\mathbf{d}), \langle Sub_1, Sub_2 \rangle)$  is a relation for  $Seq(\mathbf{d}), \langle Sub_1, Sub_2 \rangle$  satisfying Exp. (15). Then, the inside probability  $\beta$  and outside probability  $\alpha$  of the two kinds of complete sets can be calculated by Exp. (16) to Exp. (19),

<sup>10</sup>We further restrict the *root* mark  $\langle /s \rangle$  to take only one modifier (the situation when  $d_1 = l + 1$  in Exp. (7)), according to the general restrictions of the dependency grammar.

<sup>11</sup>From the restriction in Exp. (2),  $d_0$  should not be equal to  $d_1$ . This is only possible for those  $Seq(\mathbf{d})$  at the start of the recursive definition, where the word at  $d_0$  is actually a  $\langle /L \rangle$  tag or a  $\langle /R \rangle$  tag, which does not have an absolute position in a sentence.

where  $p(\mathbf{d})$  is the probability of the lexical dependency N-gram represented by  $\mathbf{d}$  in a sentence.

$$\begin{aligned} \beta(Link(\mathbf{d})) &= \\ &\sum_{\substack{\langle Sub_1, Sub_2 \rangle, \text{ s.t.} \\ R_{Link}(Link(\mathbf{d}), \langle Sub_1, Sub_2 \rangle)}} \beta(Sub_1)\beta(Sub_2)p(\mathbf{d}) \end{aligned} \quad (16)$$

$$\begin{aligned} \beta(Seq(\mathbf{d})) &= \\ &\sum_{\substack{\langle Sub_1, Sub_2 \rangle, \text{ s.t.} \\ R_{Seq}(Seq(\mathbf{d}), \langle Sub_1, Sub_2 \rangle)}} \beta(Sub_1)\beta(Sub_2) \end{aligned} \quad (17)$$

$$\begin{aligned} \alpha(Link(\mathbf{d})) &= \\ &\sum_{\substack{\langle Sup, Con \rangle, \text{ s.t.} \\ R_{Seq}(Sup, \langle Link(\mathbf{d}), Con \rangle)}} \alpha(Sup)\beta(Con) \end{aligned} \quad (18)$$

$$\begin{aligned} \alpha(Seq(\mathbf{d})) &= \\ &\sum_{\substack{\langle Sup, Con \rangle, \text{ s.t.} \\ R_{Link}(Sup, \langle Seq(\mathbf{d}), Con \rangle)}} \alpha(Sup)\beta(Con)p(\mathbf{d}') \\ &+ \sum_{\substack{\langle Sup, Con \rangle, \text{ s.t.} \\ R_{Seq}(Sup, \langle Seq(\mathbf{d}), Con \rangle)}} \alpha(Sup)\beta(Con) \end{aligned} \quad (19)$$

(where  $\mathbf{d}'$  is the N-tuple of *Sup*)

Specifically, Exp. (16) and Exp. (17) can be directly derived from the definitions of Exp. (7) and Exp. (9), respectively. Further, a complete-link set can only be a component of a complete-sequence set from Exp. (9), while a complete-sequence set can be both a component of a complete-link set from Exp. (7), and a component of a complete-sequence set from Exp. (9). As a result, Exp. (18) and Exp. (19) can be derived respectively.

For all  $Seq(\mathbf{d})$  with  $\langle /L \rangle$  or  $\langle /R \rangle$ , we use:

$$\beta(Seq(\mathbf{d})) = p(\mathbf{d}) \quad (20)$$

as the start of the calculation. At the end of the calculation, the probability of the whole sentence  $S = w_0^{l+1}$  can be obtained as:

$$P(S) = \beta(Seq(\mathbf{d} = (1, l + 1, 0, \dots, 0))) \quad (21)$$

For the re-estimation, we can get the probabilistic counts<sup>12</sup> of a dependency N-gram represented by  $\mathbf{d}$  in a sentence using:

$$\beta(Link(\mathbf{d})) \cdot \alpha(Link(\mathbf{d})) \cdot P(S)^{-1} \quad (22)$$

according to the inside-outside algorithm. Finally, all the counts of a dependency N-gram in the training corpus are added and normalized using Exp. (1), to update the model parameters.

<sup>12</sup>They are no longer integers.

<sup>13</sup>For the situation in Exp. (20), we use  $\frac{\beta(Seq(\mathbf{d})) \cdot \alpha(Seq(\mathbf{d}))}{P(S)}$ .

## 5 Experiments

### 5.1 Experiment Setting

#### Corpus

As the proposed dependency N-gram model and estimation algorithm are language-independent, we conduct experiments using four different languages: English, German, Spanish and Japanese. The corpora we use for English, German, and Spanish are the sets of sentences with 5 – 15 words from the corresponding single-language corpora of **Europarl**<sup>14</sup> (Koehn, 2005). The corpus for Japanese is the set of sentences with 5 – 20 words from the Japanese side of the **NTCIR-8** corpus (Fujii et al., 2010). We take one two-hundredth of the sentences from a corpus to form each of the development and test sets used in experiments, and the remaining sentences are used for training. The details of training, development and test sets are shown in Tables 1 and 2.

language	sentences	types	tokens
English	400,100	40,913	4,355,333
German	422,951	105,303	4,545,263
Spanish	370,791	58,314	4,007,816
Japanese	477,118	47,930	7,758,437

Table 1: The training sets.

language	development set	test set
English	2,020	2,021
German	2,136	2,136
Spanish	1,872	1,873
Japanese	2,409	2,410

Table 2: The numbers of sentences in development and test sets.

#### Parameter Collection and Initialization

In order to investigate the fundamental properties of the model and algorithm, we do not use any pruning or approximating methods in the parameter estimation. Specifically, we collect from the raw corpora all possible lexical dependency N-grams<sup>15</sup> without any cut-off thresholds for models of every order. Before estimation, we use relative frequency to initialize the probabilities.

<sup>14</sup><http://www.statmt.org/europarl/>

<sup>15</sup>As Japanese is a typical head-final language, that is, the head word always comes after its modifiers, we only take the left-oriented (from head to modifier) dependency links into account. For the other three languages, dependency links of both two orientations are considered. The parameter collection and initialization do not take the structure into account.

## 5.2 Results

### Algorithm Convergence

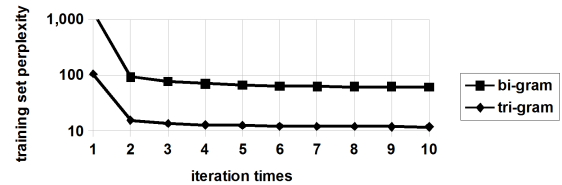


Figure 9: The English training set perplexities before each iteration. (The y-axis is logarithmic.)

Figure 9 shows the change of English training set perplexities before each iteration by the proposed estimation algorithm, for 2 (bi-) and 3 (tri-) order dependency N-gram models. The convergence trend along with the iteration times can be observed. For the dependency bi-gram, the training set perplexity becomes nearly stable after 5 iterations. However, for the dependency tri-gram, the first iteration already reaches a very low training set perplexity and it does not change much in further iterations. This phenomenon suggests that the non-pruned dependency tri-gram model may already be too complex a model with too many parameters, so the features of the training set are represented well, resulting in a low perplexity. This suggests the model is over-fitting the data. We discuss this in Sec. 5.3.

#### Test Set Perplexity

As well as the training set perplexity, the perplexity of a test set which has not been used in parameter estimation should be investigated in evaluation. Because different order dependency N-gram models are trained separately, we use linear interpolation in calculating the test set perplexity. Specifically, we use the hand-out development set to tune the interpolation coefficients (weights) and to select the iteration times of different order models to minimize the development set perplexity. Then we use the tuned weights to combine the iteration-time-selected models in the test set perplexity calculation. The reason for using simple and straightforward linear interpolation is also that we want to discover the essential aspects of the proposed model and algorithm, so we use no further smoothing approaches. As the lowest order of a dependency N-gram is two, we use a uni-gram model with modified Kneser-Ney discounting to handle the unknown words. The uni-gram model is interpolated with the dependency bi-gram model. Fur-



language	dev-ppl (bi / tri)	test-ppl (bi / tri)	$iter_{bi}$	$iter_{tri}$	$\lambda_{uni}$	$\lambda_{bi}$	$\lambda_{tri}$
English	145 / 143	159 / 156	6	1	0.93	0.99	0.13
German	268 / 256	265 / 261	12	1	0.88	0.98	0.04
Spanish	165 / 164	159 / 158	7	1	0.92	0.99	0.04
Japanese (left-only)	88 / 67	88 / 67	4	1	0.86	0.99	0.70

Table 3: The development set perplexities (dev-ppl) and test set perplexities (test-ppl) of dependency N-gram models ( $N = 2$  (bi),  $3$  (tri)). The iteration times in dependency bi- and tri-gram model training are  $iter_{bi}$  and  $iter_{tri}$ , respectively. The weights of uni-gram, dependency bi- and tri-gram models are  $\lambda_{uni}$ ,  $\lambda_{bi}$  and  $\lambda_{tri}$ .  $(1 - \lambda_{bi})$  and  $(1 - \lambda_{tri})$  are assigned to the interpolated lower order models and  $(1 - \lambda_{uni})$  is assigned to the  $\langle /L \rangle$  and  $\langle /R \rangle$  tags.

thermore, as the  $\langle /L \rangle$  tag and  $\langle /R \rangle$  tag are taken as general words but they never really appear in a training set, we treat them separately, and interpolate them with the uni-gram model.

language	MLE (bi / tri)	MKN (bi / tri)
English	162 / 457	157 / 86
German	396 / 1371	252 / 139
Spanish	176 / 499	161 / 86
Japanese	62 / 87	91 / 39

Table 4: The test set perplexities of the original N-gram models. MLE is the maximum likelihood estimation realized by setting the *adding delta* to 0 in adding smoothing. MKN is the interpolated modified Kneser-Ney discounting.

In Table 3, we show the development and test set perplexities of the linear-interpolated dependency bi- and tri-gram models. For comparison, we used **SRILM**<sup>16</sup> (Stolcke, 2002) to build two original N-gram language models on the same training sets: one is constructed by maximum likelihood estimation without any smoothing, the other one is constructed by state-of-the-art interpolated modified Kneser-Ney discounting. We calculate the test set perplexities of the two N-gram language models on the same test sets. The results are listed in Table 4. In both Table 3 and Table 4, the perplexities are calculated according to the number of lexical words, and the tags used for normalization are not counted<sup>17</sup>. We discuss these results in Sec. 5.3.

<sup>16</sup><http://www.speech.sri.com/projects/srilm/>

<sup>17</sup>That is, we do not count the  $\langle /s \rangle$  tag in the original N-gram language models, or  $\langle /L \rangle$  and  $\langle /R \rangle$  in our models. If they are included, the perplexities decrease. In the original N-gram model, this is because a  $\langle /s \rangle$  tag nearly always appears after the period mark. The effect is even more dramatic in our model, as each word in a sentence has a  $\langle /L \rangle$  and a  $\langle /R \rangle$  tag to normalize its modifier numbers, so the token number in a sentence is multiplied by. Therefore, we only count the lexical words in perplexity calculation for fairness.

### 5.3 Discussion

#### Parameter Number

For a sentence with  $l$  words, the number of dependency N-gram that can be collected increases exponentially as  $O(l^N)$  if we consider all possible combinations. Although for a given  $N$ , the proposed algorithm takes a time which is polynomial in the sentence length  $l$ , a large  $N$  will be practically intractable, especially for long sentences. In Fig. 10, we show the numbers of complete sets of different order dependency N-gram models for different sentence lengths.

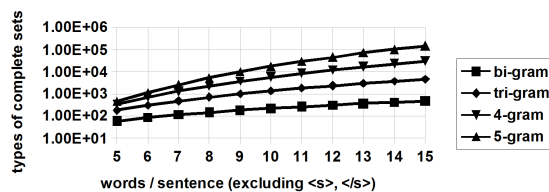


Figure 10: The numbers of complete sets. (The y-axis is logarithmic.)

This behavior is also related to the over-fitting problem because our algorithm is essentially an iterative maximum likelihood estimation. A model that is too complex will be too specific to the training set. From Table 3, we see that the performance of a dependency tri-gram model will saturate after only one iteration, which is also indicated in Fig. 9, and does little to improve the test set perplexities. The exception is Japanese, where the dependency tri-gram does improve the performance. The linguistic reason for this is that Japanese is a head-final language with a simpler syntactic structure, so we restrict the dependency link in Japanese to “left only”, which leads to a model with fewer parameters. Consequently, the high order model performs better. From the experimental results, we can see that the proposed algorithm has the usual strengths and weaknesses of an EM algorithm.

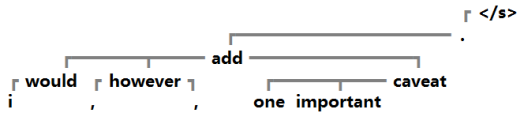


Figure 11: The best dependency structure of the English sentence “*i would , however , add one important caveat .*”

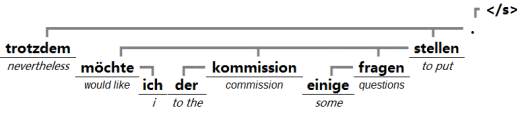


Figure 12: The best dependency structure of the German sentence “*trotzdem möchte ich der kommission einige fragen stellen .*”

### Test Set Perplexity

Comparing the test set perplexities in Tables 3 and 4, we can see the dependency bi-gram model achieves the same, or sometimes better performance of the original N-gram language models. However, when we look at the tri-grams, the interpolated modified Kneser-Ney (KN) discounting method, which is state-of-the-art, shows its strength and our dependency model does not produce much improvement for the reasons we described above<sup>18</sup>. As the modified KN method uses an efficient discounting to avoid the over-fitting problem, and our model has no smoothing, the difference in performance is reasonable for complex models. On the other hand, the generally competitive results of our bi-gram model and its performance on Japanese show that our model is a promising one, particularly if the number of parameters can be reduced.

### Model Preference

In Fig. 11 to Fig. 14, we present examples of the best dependency structures generated by our approach of sentences in test sets. We used the settings in Table 3 and generated them by the Viterbi algorithm (Viterbi, 1967). It can be seen the proposed approach can reveal features of specific languages even though it is unsupervised: e.g. the final-position verb “*stellen*” and its relation with the second-position auxiliary verb “*möchte*” in the German sentence. The results also show a preference for associating semantic relations and making the function words<sup>19</sup> of a language the modifiers of the content words. For example, in the Spanish sentence, syntactically the preposition “*a*”

<sup>18</sup>For Japanese, the result is improved by the dependency tri-gram model, but the original tri-gram model with interpolated modified KN discounting method performs much better.

<sup>19</sup>Articles, prepositions, etc.

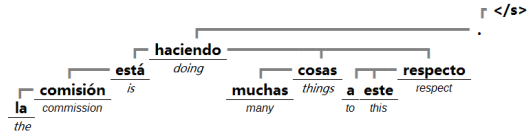


Figure 13: The best dependency structure of the Spanish sentence “*la comision está haciendo muchas cosas a este respecto .*”

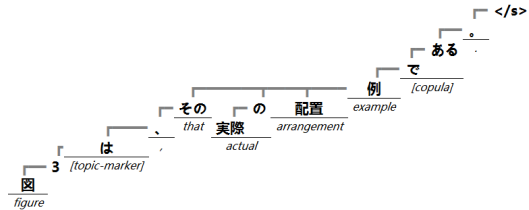


Figure 14: The best dependency structure of the Japanese sentence “*図 3 は、その実際の配置例である。*”

is the head of the noun “*respecto*”, but in unsupervised training, our model prefers to assign “*a*” to be the modifier of “*respecto*” and directly link two content words: “*respecto*” and the verb “*haciendo*”. We think this is because the probabilities of  $\langle /L \rangle$  and  $\langle /R \rangle$  tags have large estimates, especially when they appear after function words, which prevents them from having modifiers<sup>20</sup>.

## 6 Conclusion and Future Work

In this paper, we proposed a generative dependency N-gram language model and the definition of the complete sets for arbitrary order, by which an unsupervised parameter estimation algorithm is facilitated. The experimental results demonstrate the applicability and the properties of the proposed approach. In future work, we will develop methods of parameter pruning and discounting to handle the over-fitting problem. As the the proposed dependency language model is intrinsically complex, we also plan to do more fundamental simplifications. On the other hand, although our proposed algorithm is unsupervised, the output of a trained parser, which can provide clear and lexical heuristics, can be integrated in it. We will investigate this possibility and evaluate the performance by linguistically motivated criteria.

### Acknowledgment

We would like to thank anonymous reviewers for their valuable comments and suggestions. This work was supported by JSPS KAKENHI Grant Number 24650063.

<sup>20</sup>This tendency, however, is correct for articles, such as the “*der*” in German and “*la*” in Spanish.



## References

- Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vicent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.
- Ciprian Chelba and Frederick Jelinek. 2000. Structured language modeling. *Computer Speech and Language*, 14(4):283–332.
- Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, TR-10-98, Computer Science Group, Harvard Univ.
- Michael Collins. 1998. Three generative, lexicalised models for statistical parsing. In *Proc. of ACL 1998*, pages 16–23.
- Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: an exploration. In *Proc. of COLING 1996*, pages 340–345.
- Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizenya, and Sayori Shimohata. 2010. Overview of the patent translation task at the NTCIR-8 workshop. In *Proc. of NTCIR-8*, pages 371–376.
- Jianfeng Gao and Hisami Suzuki. 2003. Unsupervised learning of dependency structure for language modeling. In *Proc. of ACL 2003*, pages 521–580.
- Yvette Graham and Josef van Genabith. 2010. Deep syntax language models and statistical machine translation. In *Proc. of SSST at COLING 2010*, pages 118–126.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL 2003*, pages 423–430.
- Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proc. of ACL 2004*, pages 478–485.
- Philipp Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *Proc. of MT summit 2005*, pages 79–86.
- Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc. of COLING 2002*, pages 1–7.
- K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4(1):35–56.
- Seungmi Lee and Key-Sun Choi. 1997. Reestimation and best-first parsing algorithm for probabilistic dependency grammars. In *Proc. of WVLC 1997*, pages 41–55.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.
- Mark A. Paskin. 2002. Grammatical digrams. *Advances In Neural Information Processing Systems 14*, 1:91–97.
- Andreas Stolcke, Ciprian Chelba, David Engle, Victor Jimenez, Lidia Mangu, Harry Printz, Eric Ristad, Roni Rosenfeld, Dekai Wu, Frederick Jelinek, and Sanjeev Khudanpur. 1997. Dependency language modeling.
- Andreas Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *Proc. of ICSLP 2002*, pages 901–904.
- Christoph Tillmann and Hermann Ney. 1997. Word triggers and the EM algorithm. In *Proc. of CoNLL 1997*, pages 117–124.
- Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on Information Theory*, 13(2):260–269.

# Learning a Product of Experts with Elitist Lasso

**Mengqiu Wang**

Computer Science Department  
Stanford University  
Stanford, CA 94305, USA  
mengqiu@cs.stanford.edu

**Christopher D. Manning**

Computer Science Department  
Stanford University  
Stanford, CA 94305, USA  
manning@cs.stanford.edu

## Abstract

Discriminative models such as logistic regression profit from the ability to incorporate arbitrary rich features; however, complex dependencies among overlapping features can often result in weight undertraining. One popular method that attempts to mitigate this problem is logarithmic opinion pools (LOP), which is a specialized form of product of experts model that automatically adjusts the weighting among experts. A major problem with LOP is that it requires significant amounts of domain expertise in designing effective experts. We propose a novel method that learns to induce experts — not just the weighting between them — through the use of a mixed  $\ell_2\ell_1$  norm as previously seen in *elitist lasso*. Unlike its more popular sibling  $\ell_1\ell_2$  norm (used in group lasso), which seeks feature sparsity at the group-level,  $\ell_2\ell_1$  norm encourages sparsity within feature groups. We demonstrate how this property can be leveraged as a competition mechanism to induce groups of diverse experts, and introduce a new formulation of elitist lasso MaxEnt in the FOBOS optimization framework (Duchi and Singer, 2009). Results on Named Entity Recognition task suggest that this method gives consistent improvements over a standard logistic regression model, and is more effective than conventional induction schemes for experts.

## 1 Introduction

Conditionally trained discriminative models like logistic regression (a.k.a., MaxEnt models) gain

power from their ability to incorporate a large number of arbitrarily overlapping features. But such models also exhibit complex feature dependencies and therefore are susceptible to the problem of weight undertraining — the contributions of certain features are overlooked because of features they co-occur with (Sutton et al., 2007; Hinton et al., 2012).

One popular method that attempts to mitigate this problem is logarithmic opinion pools (LOP) (Heskes, 1998; Smith et al., 2005; Sutton et al., 2007). LOP works in a similar fashion to a product of experts model, in which a number of experts are individually assembled first, and then their predictions are combined multiplicatively to create an ensemble model. Experts are generated by first partitioning the feature space into subsets, then an independent model (expert) is trained on each subset of features (Sutton et al., 2007). Under the assumption that strongly correlated features are partitioned into separate subsets, this method effectively forces the models to learn how to make predictions with each subset of the features independently. It also helps in scenarios where some strong features stop other features from getting weight, a problem known as feature co-adaptation. The theoretical justification for favoring a product ensemble over an additive ensemble is that inference with a product of log-linear models is significantly easier than inference with a sum. Sutton et al. (2007) also suggests that product ensembles give better results than additive mixtures on sequence labeling tasks.

Smith et al. (2005) and Sutton et al. (2007) showed that the quality of experts and diversity among them have a direct impact on the final performance of the ensemble. Designing effective and diverse experts was left as an art, and requires

a great deal of domain expertise.

In this paper, we directly learn to induce diverse experts by using a mixed  $\ell_2\ell_1$  norm known as *elitist lasso* in the context of generalized linear models (Kowalski and Torr esani, 2009). We design a novel competition mechanism to encourage experts to use non-overlapping feature sets, effectively learning a partition of the feature space. Efficient optimization of a maximum conditional likelihood objective with respect to  $\ell_2\ell_1$  norm is non-trivial and has not been previously studied. We propose a novel formulation to incorporate  $\ell_2\ell_1$  norm in the FOBOS optimization framework (Duchi and Singer, 2009). Experiments on Named Entity Recognition task suggest that our proposed method gives consistent improvements over a baseline MaxEnt model and conventional LOP experts. In particular, our method gives the most improvements in recognizing entity types for out-of-vocabulary words.

## 2 MaxEnt Model

Given an input sequence  $\mathbf{x}$  (e.g., words in a sentence), a MaxEnt model defines the conditional probability of an output variable  $y$  (e.g., named-entity tag of a word) as follows:

$$P(y = v_i | \mathbf{x}) = \frac{\exp\left(\sum_{k=1}^K \theta_k(v_i) f_k(\mathbf{x})\right)}{\sum_{j=1}^V \exp\left(\sum_{k=1}^K \theta_k(v_j) f_k(\mathbf{x})\right)}$$

where  $f_k(\mathbf{x})$  is the  $k$ th input feature,  $K$  is the total number of input features, and  $\theta_k(y)$  is the weight parameter associated with feature  $f_k$  for output class  $v_i$ . We denote the output classes of  $y$  to be  $\{v_1, v_2, \dots, v_V\}$ , where  $V$  is the total number of output classes.

It is helpful to consider the connection between a MaxEnt model and a Linear Network model commonly seen in the neural network literature, by re-expressing this objective using a *softmax* activation function. The softmax function is defined as:

$$\text{softmax}(q(y)) = \frac{\exp(q(y))}{\sum_{y'} \exp(q(y'))}$$

in our case  $q(y) = \sum_{k=1}^K \theta_k(y) f_k(\mathbf{x})$ , and we have:

$$P(y | \mathbf{x}) = \text{softmax}\left(\sum_{k=1}^K \theta_k(y) f_k(\mathbf{x})\right)$$

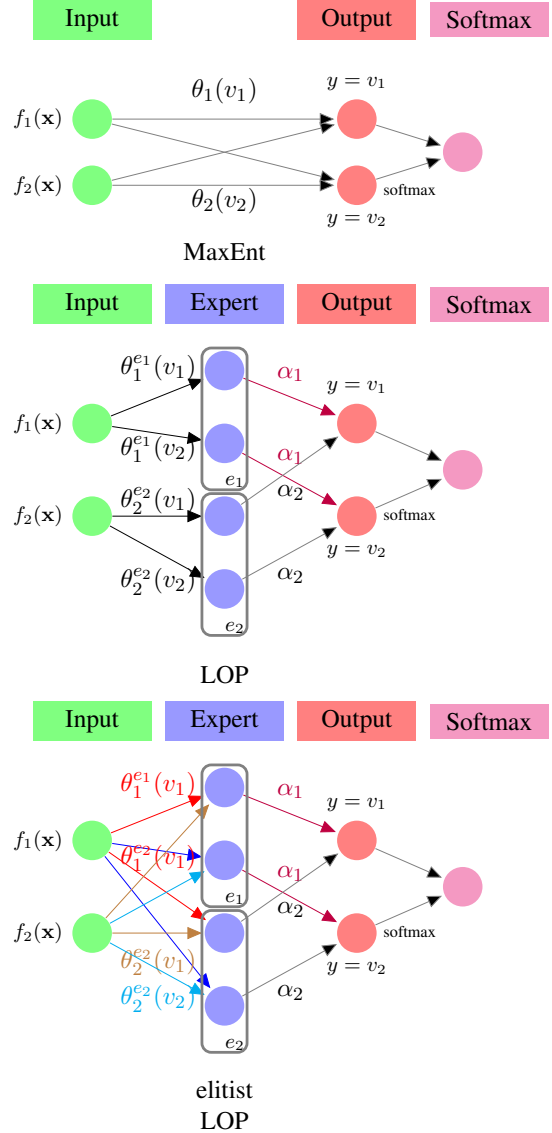


Figure 1: The top part shows a regular MaxEnt model with 2 features and 2 output classes. The middle part shows a LOP with two experts.  $\alpha_1$  and  $\alpha_2$  are the expert mixing coefficients to be learned in the LOP model. The bottom part shows an elitist LOP expert induction model with two experts. For each feature weight parameter  $\theta$ , the superscript denotes which feature group it belongs to (e.g.,  $\theta^{e_1}$  belongs to expert  $e_1$ ); the subscript denotes which input feature it belongs to (e.g.,  $\theta_1$  means it applies to input feature  $f_1(\mathbf{x})$ ); and the value in parentheses denotes which output class this feature is associated with (e.g.,  $\theta(v_1)$  applies to output class  $y = v_1$ ). Features that belong to the same group are shown in the same color.

We visualize this using a neural network style diagram in the top part of Figure 1. In this diagram, the value at each node is computed as the sum of all incoming edge weights (parameters in the model) multiplied by the values of nodes on the originating end of edges. In this example,  $K = 2$  and we have two input features ( $f_1(\mathbf{x})$  and  $f_2(\mathbf{x})$ ); there are two possible value assignments for  $y$  ( $v_1$  and  $v_2$ ). Correspondingly, there are four weight parameters ( $\theta_i(v_j)$ , for  $i = 1, 2, j = 1, 2$ ) as shown by the directed edge going from “Input” layer nodes to “Output” layer nodes. We apply a softmax function to the “Output” layer to produce the probability assignment of each output class.

### 3 Weight Undertraining

The problem of weight undertraining occurs in discriminatively trained classifiers when some strongly indicative features occur during training but not at test time. Because of the presence of such strong features during training, weaker features that occur at both training and test time do not receive enough weight, and thus impede the classifier from reaching its full potential at test time. This phenomenon is also known as feature co-adaptation (Hinton et al., 2012). Sutton et al. (2007) gave an excellent demonstration of this problem using a synthetic logistic regression model. In their example, the output function is the softmax of the linear sum of a set of weights  $x_1$  to  $x_n$ , where each of the  $x_i$  acts as a “weak” predictor. A “strong” feature which takes the form of  $x' = \sum_{i=1}^n x_i + \mathcal{N}$  (where  $\mathcal{N}$  is an added Gaussian noise) is then added to the classifier during training but removed at test time, which simulates a feature co-adaptation scenario. Simulation results show that classifier accuracy can drop by as much as 40% as a result of weight undertraining.

This problem is difficult to combat because we do not know a priori what set of features will be present at test time. However, a number of methods including feature bagging (Sutton et al., 2007), random feature dropout (Hinton et al., 2012), and logarithmic opinion pools (Heskes, 1998; Smith et al., 2005) have been introduced as ways to alleviate weight undertraining.

### 4 Logarithmic Opinion Pools

The idea behind logarithmic opinion pools (LOP) is that instead of training one classifier, we can train a set of classifiers with non-overlapping or

competing feature sets. At test time, we can combine these classifiers and average their results. By creating a diverse set of classifiers and training each of them separately, we can potentially reduce the effect of weight undertraining. For example, when two strongly correlated features are partitioned into different classifiers, they do not overshadow each other. Unlike voting or feature bagging where an additive ensemble of experts is used, LOP derives a joint model by taking a product of experts.<sup>1</sup>

The conditional probability of a LOP ensemble can be expressed as:

$$P_{LOP}(y|\mathbf{x}) = \frac{\prod_{j=1}^n [P_j(y|\mathbf{x})]^{\alpha_j}}{\sum_{y'} \prod_{j=1}^n [P_j(y'|\mathbf{x})]^{\alpha_j}}$$

$$P_j(y|\mathbf{x}) = \text{softmax} \left( \sum_{k=1}^K \theta_k^{e_j}(y) f_k(\mathbf{x}) \right)$$

where  $n$  is the total number of experts,  $P_j(y|\mathbf{x})$  is the probability assignment of expert  $j$ , and  $\alpha_j$  is the mixing coefficient for this expert.

The experts are typically selected so that they capture different signals from the training data (e.g., via feature subsetting). Each expert is trained separately, and their model parameters  $\theta_k^{e_j}(y)$  are fixed during LOP combination.

Early work on LOPs either fixed the expert mixing coefficients to some uniformly assigned value (Hinton, 1999), or used some arbitrary hand-picked values (Sutton et al., 2007). In both of these two cases, no new parameter values need to be learned, therefore no extra training is required.

Smith et al. (2005) proposed to learn the weights by maximizing log-likelihood of the ensemble product model, and showed that the learned weights work better than uniformly assigned values.

We can visualize the LOP again using a network diagram, as shown in the middle part of Figure 1. Two experts are shown in this diagram, each represented by a plate in the “Expert” layer. This example illustrates an expert generation scheme by partitioning the features into non-overlapping subsets – expert  $e_1$  has feature function  $f_1(\mathbf{x})$  while expert  $e_2$  has feature function  $f_2(\mathbf{x})$ . The weights of the two experts are pre-trained and remain fixed. Training the LOP model involves only learning the  $\alpha$  mixing parameters.

<sup>1</sup>A product in the raw probability space is equivalent to a sum in the logarithmic space, and hence the name “logarithmic” opinion pools.

## 5 Automatic Induction of Experts

A major concern in adopting existing LOP methods is that there are no straightforward guidelines on how to create experts. Different feature partition schemes can result in dramatically different performance (Smith et al., 2005). A successfully designed LOP expert ensemble typically involves non-trivial engineering effort and a significant amount of domain expertise. Furthermore, the improvements shown in one application domain by a well-designed feature partition scheme do not necessarily carry over to other domains.

Therefore, it is our goal to investigate and search for effective methods to automatically induce LOP experts. We would like to find a set of experts that are diverse (i.e., having non-overlapping feature sets) and accurate. The main idea here is that we can train multiple copies of MaxEnt models together to optimize training likelihood, but at the same time we encourage the MaxEnt models to differ as much as possible in their choice of features to employ. To meet this end, we use a  $\ell_2\ell_1$  mixed norm to regularize a LOP, and denote the new model as *elitist LOP*.

### 5.1 Elitist LOP

Before we introduce the  $\ell_2\ell_1$  norm, which has not been frequently used in NLP research, let us revisit a closely related but much better known sibling – the  $\ell_1\ell_2$  norm used in group lasso.

For a set of model parameters  $\theta$ , assume we have some feature grouping that partitions the features into  $G$  sub-groups. For simplicity, let us further assume that each feature group has the same number of features ( $M$ ). We denote the index of each feature by  $g$  (for “group”) and  $m$  (for “member”), and use  $\theta_{\hat{g}}$  to denote the feature group of index  $g$ .

The  $\ell_1\ell_2$  mixed norm used in group lasso takes the following form:

$$\begin{aligned} \|\theta\|_{1,2} &= \left( \sum_{g=1}^G \left( \sum_{m=1}^M |\theta_{g,m}|^2 \right) \right)^{1/2} \\ &= \|(\|\theta_{\hat{1}}\|_2, \dots, \|\theta_{\hat{G}}\|_2)\|_1 \end{aligned}$$

This norm applies  $\ell_2$  regularization to each group of features, but enforces  $\ell_1$  regularization at the group level. Since  $\ell_1$  norm encourages sparse solutions, therefore the effect of the mixed norm is to sparsify features at the group level by eliminating a whole group of features at a time.

It is easy to see how it relates to the  $\ell_2\ell_1$  mixed norm, which is given as:

$$\begin{aligned} \|\theta\|_{2,1} &= \left( \sum_{g=1}^G \left( \sum_{m=1}^M |\theta_{g,m}| \right) \right)^2 \Big)^{1/2} \\ &= \|(\|\theta_{\hat{1}}\|_1, \dots, \|\theta_{\hat{G}}\|_1)\|_2 \end{aligned}$$

Like  $\ell_1\ell_2$ , this is also a group-sensitive sparsity-inducing norm, but instead of encouraging sparsity across feature groups, it promotes sparsity within each feature group.

We illustrate how we apply  $\ell_2\ell_1$  norm to induce diverse experts in the bottom diagram in Figure 1. Similar to the case of LOP, we create two “Expert” layer plates, one for each expert ( $e_1$  and  $e_2$ ). But unlike traditional LOP, where each expert only gets a subset of the input features (e.g.,  $f_1(\mathbf{x}) \mapsto e_1$  and  $f_2(\mathbf{x}) \mapsto e_2$ ), each input feature is fully connected to each expert plate (i.e., there are four edges with the same expert superscript from the “Input” layer going into each plate in the “Expert” layer, shown in different colors).

We group every pair of feature weights that originate from the same input feature and end at the same output class into a separate feature group (e.g.,  $\theta_1^{e_1}(v_1)$  and  $\theta_1^{e_2}(v_1)$  belongs to one group, while  $\theta_2^{e_1}(v_1)$  and  $\theta_2^{e_2}(v_1)$  belongs to another group, etc.). In total we have four feature groups, each shown in a different color.

When we apply  $\ell_2\ell_1$  norm to this feature grouping while learning the parameter weights in the LOP model, the regularization will push most of the features that belong to the same group towards 0. But since we are optimizing a likelihood objective, if a particular input feature is indeed predictive, the model will also try to assign some weights to the weight parameters associated with this feature. These two opposite forces create a novel competition mechanism among features that belong to the same group.

This competition mechanism encourages the experts to use different sets of non-overlapping features, thus achieving the diversity effect we want. But because we are also optimizing the likelihood of the ensemble, each expert is encouraged to use parameters that are as predictive as possible. Learning all the experts together can also be seen as a form of *feature sharing*, as suggested by Ando and Zhang (2005) for multi-task learning.

## 5.2 FOBOS with $\ell_2\ell_1$ norm

In our automatic expert induction model with  $\ell_2\ell_1$  norm, the overall objective is given as:

$$\mathcal{L} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left( -\log \left( \frac{\prod_{j=1}^n [P_j(y|\mathbf{x})]^{\alpha_j}}{\sum_{y'} \prod_{j=1}^n [P_j(y|\mathbf{x})]^{\alpha_j}} \right) + \lambda \|(\|\boldsymbol{\theta}_{\hat{1}}\|_1, \dots, \|\boldsymbol{\theta}_{\hat{G}}\|_1)\|_2 \right)$$

where  $\lambda$  is a parameter that controls the regularization strength.

The feature groups are given as:

$$\begin{aligned} \boldsymbol{\theta}_{\hat{1}} &= \{\theta_1^{e_1}(v_1), \theta_1^{e_2}(v_1), \dots, \theta_1^{e_n}(v_1)\} \\ \boldsymbol{\theta}_{\hat{2}} &= \{\theta_2^{e_1}(v_1), \theta_2^{e_2}(v_1), \dots, \theta_2^{e_n}(v_1)\} \\ &\dots \\ \boldsymbol{\theta}_{\hat{K}} &= \{\theta_K^{e_1}(v_1), \theta_K^{e_2}(v_1), \dots, \theta_K^{e_n}(v_1)\} \\ \boldsymbol{\theta}_{\hat{K+1}} &= \{\theta_1^{e_1}(v_2), \theta_1^{e_2}(v_2), \dots, \theta_1^{e_n}(v_2)\} \\ \boldsymbol{\theta}_{\hat{K+2}} &= \{\theta_2^{e_1}(v_2), \theta_2^{e_2}(v_2), \dots, \theta_2^{e_n}(v_2)\} \\ &\dots \\ \boldsymbol{\theta}_{\hat{G}} &= \{\theta_K^{e_1}(v_V), \theta_K^{e_2}(v_V), \dots, \theta_K^{e_n}(v_V)\} \end{aligned}$$

There is a total of  $G = K \times V$  feature groups, and each feature group has  $M = n$  features.

The key difference here from LOP is that the  $\boldsymbol{\theta}$  parameters are not pre-trained and fixed, but jointly learned with respect to  $\ell_2\ell_1$  regularization. In the LOP case, learning the mixing coefficients  $\alpha_i$  ( $i = \{1, \dots, n\}$ ) can be done by taking the gradients of  $\alpha_i$  with respect to the training objective and directly plugging it into a gradient-based optimization framework, such as the *limited memory variable metric* (LMVM) used by Smith et al. (2005) or Stochastic Gradient Descent. However, learning the  $\boldsymbol{\theta}$  parameters in our case is not as straightforward, since the objective is non-differentiable at many points due to the  $\ell_2\ell_1$  norm. We cannot simply throw in an off-the-shelf gradient-based optimizer.

To alleviate the problem of non-differentiability, the recently proposed FOrward-Backward Splitting (FOBOS) optimization framework (Duchi and Singer, 2009) comes to mind. FOBOS is a (sub-)gradient-based framework that solves the optimization problem iteratively in two steps: in each iteration, we first take a sub-gradient step, and then take an analytical minimization step that incorporates the regularization term, which can often be solved with a closed form solution. Formally, each iteration at timestamp  $t$  in FOBOS

consists of the following two steps:

$$\begin{aligned} \boldsymbol{\theta}_{t+\frac{1}{2}} &= \boldsymbol{\theta}_t - \eta_t \mathbf{g}_t & (1) \\ \boldsymbol{\theta}_{t+1} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_{t+\frac{1}{2}}\|^2 + \eta_t \varphi(\boldsymbol{\theta}) \right\} & (2) \end{aligned}$$

$\mathbf{g}_t$  is the subgradient of  $-\log(P_{LOP}(y|\mathbf{x}))$  at  $\boldsymbol{\theta}_t$ . Step (1) is just a regular gradient-based step, where  $\eta_t$  is the step size and  $\mathbf{g}_t$  is the sub-gradient vector of parameter vector  $\boldsymbol{\theta}$ . Step (2) finds a new vector that stays close to the interim vector after step (1), but also has a low penalty score according to the regularization term  $\varphi(\boldsymbol{\theta})$  (in our case,  $\varphi(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{2,1}$ ). In the case of  $\ell_1$  and  $\ell_2$  regularization, each feature  $\theta_i$  is independent in step (2), and thus can be solved separately.<sup>2</sup>

Duchi and Singer (2009) gave closed form solutions for solving the minimization problem in step (2) for  $\ell_1$ ,  $\ell_2$ , and  $\ell_1\ell_2$  norms, but no past research has demonstrated how to solve it for  $\ell_2\ell_1$  norm. We leverage the results given by Kowalski and Torr sani (2009) for elitist lasso, and show that parameter  $\theta_{t+1,g,m}$  (the  $m^{\text{th}}$  feature of group  $g$  at timestamp  $t+1$ ) can be solved analytically as follows:

$$\begin{aligned} \theta_{t+1,g,m} &= \operatorname{sign}(\theta_{t+\frac{1}{2},g,m}) \left( |\theta_{t+\frac{1}{2},g,m}| - \tau_g \right)^+ \\ \tau_g &= \frac{\lambda'}{1 + \lambda' M_{\hat{g}}(\lambda')} \left\| x_{t+\frac{1}{2},g,1:M_{\hat{g}}(\lambda')} \right\|_1 \end{aligned}$$

$\operatorname{sign}(x)$  is the mathematical sign function. For  $x \in \mathcal{R}$ , we have  $x^+ = x$  if  $x \geq 0$  and  $x^+ = 0$  if  $x \leq 0$ .  $\lambda' = \eta_t \lambda$  is the regularization weight adjusted by the current step size.

Let  $\bar{\theta}_{\hat{g}}$  denote a new vector that holds the positive absolute value of  $\boldsymbol{\theta}_{\hat{g}}$ , sorted in descending order, i.e.,  $\bar{\theta}_{g,1} \geq \bar{\theta}_{g,2} \geq \dots \bar{\theta}_{g,M}$ .  $M_{\hat{g}}(\lambda')$  is a positive integer given by the following definition:

$$\bar{\theta}_{g,M_{\hat{g}}(\lambda')+1} \leq \sum_{m=1}^{M_{\hat{g}}(\lambda')+1} \left( \bar{\theta}_{g,m} - \bar{\theta}_{g,M_{\hat{g}}(\lambda')+1} \right)$$

and

$$\bar{\theta}_{g,M_{\hat{g}}(\lambda')} > \lambda' \sum_{m=1}^{M_{\hat{g}}(\lambda')} \left( \bar{\theta}_{g,m} - \bar{\theta}_{g,M_{\hat{g}}(\lambda')} \right)$$

<sup>2</sup>This property turns out to be of great importance in real-world large scale scenarios, since it allows the optimization of high-dimensional feature vectors to be parallelized.

$x_{t+\frac{1}{2},g,1:M_{\hat{g}}(\lambda')}$  is then the subset of features from 1 to  $M_{\hat{g}}(\lambda')$  – in descending order of their absolute value – in group  $g$  at timestamp  $t + \frac{1}{2}$ .

One problem with finding the exact solution of elitist lasso as illustrated above is that computing the value  $M_{\hat{g}}(\lambda')$  involves sorting the parameter vector  $\theta$ , which is a  $O(n \log n)$  operation. When the number of features within each group is large, this could be prohibitively expensive. An approximate solution simply replaces  $M_{\hat{g}}(\lambda')$  with the size of the group  $M$ . Kowalski and Torr sani (2009) found that approximated elitist lasso works nearly as well as the exact version, but is faster and easier to implement. We adopt this approximation in our experiments.

With this, we have described a general-purpose method for solving any convex optimization problem with  $\ell_2\ell_1$  norm.

## 6 Experiments

We evaluate the performance of our proposed method on the task of Named Entity Recognition (NER). The first dataset we evaluate on is the standard CoNLL-2003 English shared task benchmark dataset (Sang and Meulder, 2003), which is a collection of documents from Reuters newswire articles, annotated with four entity types: *Person*, *Location*, *Organization*, and *Miscellaneous*. We adopt the BIOES-style annotation standard. Beginning and intermediate positions of an entity are marked with *B*- and *I*- tags, and non-entities with *O* tag. The training set contains 204K words (14K sentences), the development set contains 51K words (3.3K sentences), and the test set contains 46K words (3.5K sentences). The second dataset is the MUC6/7 set, which contains 255K words for training and 59K words for testing. The MUC data is annotated with 7 entity types. It is missing the *Miscellaneous* entity type, but includes 4 additional entity types that do not occur in CoNLL-2003: *Date*, *Time*, *Money*, and *Percent*.

We used a comprehensive set of features from Finkel et al. (2005) for training the MaxEnt model. A total number of 437905 features were generated on the CoNLL-2003 training dataset. The most important features are listed in Table 1.

### 6.1 Experimental Setup

We tuned the regularization parameters in the  $\ell_1$ ,  $\ell_2$  and  $\ell_1\ell_2$  norms on the development dataset via grid search. We used a tuning procedure similar to

<ul style="list-style-type: none"> <li>– The word, word shape (e.g., whether capitalized, numeric, etc.), and letter n-grams (up to 6gram) at current position</li> <li>– The word and word shape of the previous and next position</li> <li>– Previous word shape in conjunction with current word shape</li> <li>– Disjunctive word set of the previous and next 4 positions</li> <li>– Capitalization pattern in a 3 word window</li> <li>– The current word matched against a list of name titles (e.g., Mr., Mrs.)</li> <li>– Previous two words in conjunction with the word shape of the previous word</li> </ul>
--

Table 1: MaxEnt features.

the one used in Turian et al. (2010) where we evaluate results on the development set after each optimization iteration, and terminate the procedure after not observing a performance increase on the development set in 25 continuous iterations. We found 150 to be a good  $\lambda$  value for  $\ell_2$ , 0.1 for  $\ell_1$  and 0.1 for  $\ell_2\ell_1$ . Similarly, we tuned the number of experts for both LOP baselines and the elitist LOP induction scheme. All model parameters were initialized to a random value in  $[-0.1, 0.1]$ .

Results are measured in precision (P), recall (R) and F<sub>1</sub> score. Statistical significance is measured using the paired bootstrap resampling method (Efron and Tibshirani, 1993). We compare our results against a MaxEnt baseline model with both  $\ell_1$  and  $\ell_2$  regularization, as well as an automatic expert induction model with  $\ell_1$  norm.

Two commonly used LOP expert induction schemes were also compared. In the first scheme (*LOP random*), experts are constructed by randomly partitioning the features into  $n$  subsets, where  $n$  is the number of experts. Each expert therefore has  $\frac{1}{n}$ th of the full feature set. The second scheme (*LOP sequential*) works in a similar way, but instead of random partition, we create feature subsets sequentially and preserve the relative order of the features. We also list results reported in Smith et al. (2005) on the same dataset for comparison. They experimented with three manually crafted expert induction schemes (*Smith et al. 05 {simple;positional;label}*) for LOP with a Conditional Random Field (CRF), which is a more powerful sequence model than our MaxEnt baseline.

### 6.2 Main Results

Results on the CoNLL and MUC dataset are shown in Table 2. The proposed automatic expert induction scheme (row *elitist LOP*) gives consistent and statistically significant improvements over the MaxEnt baselines on both CoNLL and MUC test sets.

In particular, on the CoNLL test set, we ob-

	Models	# of experts	Dev Set			Test Set		
			Precision	Recall	F <sub>1</sub>	Precision	Recall	F <sub>1</sub>
CoNLL	MaxEnt- $\ell_2$	1	88.46	87.73	88.09	81.42	81.64	81.53
	MaxEnt- $\ell_1$	1	88.46	89.63	89.04	82.20	84.24	83.21 <sup>†</sup>
	LOP random	2	88.75	90.79	89.76	82.73	85.84	84.25 <sup>†‡</sup>
	LOP sequential	2	88.68	90.79	89.72	83.03	86.03	84.50 <sup>†‡</sup>
	LOP- $\ell_1$	5	88.76	90.41	89.58	82.90	85.94	84.40 <sup>†‡</sup>
	elitist LOP	3	89.65	91.08	90.36	83.96	86.67	85.29 <sup>†‡</sup>
	Smith et al. 05 simple	2	-	-	90.26	-	-	84.22
	Smith et al. 05 positional	3	-	-	90.35	-	-	84.71
	Smith et al. 05 label	5	-	-	89.30	-	-	83.27
MUC	MaxEnt- $\ell_2$	1	91.47	86.20	88.76	89.06	80.44	84.53
	MaxEnt- $\ell_1$	1	92.56	85.50	88.89	89.61	79.09	84.02
	LOP random	2	93.77	84.95	89.14	91.05	78.89	84.54
	LOP sequential	2	94.34	84.18	88.97	91.71	77.42	83.96
	LOP- $\ell_1$	5	93.25	86.09	89.53	90.70	79.57	84.78 <sup>‡</sup>
	elitist LOP	3	93.47	86.48	89.84	91.22	80.15	85.33 <sup>†‡</sup>

Table 2: Results of NER on CoNLL and MUC dataset. <sup>†</sup> and <sup>‡</sup> indicate statistically significantly better F<sub>1</sub> scores than the MaxEnt- $\ell_2$  and MaxEnt- $\ell_1$  baselines, respectively.

	Models	IV			OOV		
		Precision	Recall	F <sub>1</sub>	Precision	Recall	F <sub>1</sub>
CoNLL	MaxEnt- $\ell_2$	89.21	86.81	88.00	85.69	80.61	83.08
	elitist LOP	90.25	89.85	90.05 <sup>†</sup>	86.30	86.67	86.49 <sup>†</sup>
MUC	MaxEnt- $\ell_2$	90.21	80.88	85.29	84.90	78.80	81.74
	elitist LOP	92.27	80.02	85.71	87.49	80.62	83.91 <sup>†</sup>

Table 3: OOV and IV results breakdown. <sup>†</sup> indicates statistically significantly better F<sub>1</sub> scores than the MaxEnt- $\ell_2$  baseline.

serve an improvement of 3.7% absolute F<sub>1</sub> over *MaxEnt- $\ell_2$* . The gains are particularly large in recall (by 5% absolute score), although there is also a 2.5% improvement in precision. The two conventional LOP induction schemes also show significant improvements over the MaxEnt baselines on this dataset, with the sequential feature partition scheme working slightly better than the random feature partition. However, elitist LOP outperforms the two LOP schemes by as much as 1% in absolute F<sub>1</sub>, and also gives better results than manually crafted experts for CRFs in Smith et al. (2005).

On the MUC test set, while elitist LOP still outperforms the MaxEnt baselines, the LOP schemes do not help or in some cases hurt performance. This suggests the lack of robustness of LOP which was discussed in Section 5. The automatically learned LOP experts are more robust on this data set, and gives a 0.8% improvement measured in absolute F<sub>1</sub> score over the *MaxEnt- $\ell_2$*  baseline.

The elitist LOP model is more expressive than MaxEnt since it has more parameters to be combined. To understand whether performance gain is obtained by the expressiveness or by the regularization of  $\ell_2\ell_1$  norm, we compare against an elitist

LOP model with just  $\ell_1$  regularization (*LOP- $\ell_1$* ). Results show that the  $\ell_2\ell_1$  norm is a better regularizer for avoiding overfitting with these models. We also see a significant gain in *LOP- $\ell_1$*  over *MaxEnt- $\ell_1$* , suggesting that the expressiveness of the model also helps.

### 6.3 Out-of-vocabulary vs. In-vocabulary

To further understand *how* our method improves over the MaxEnt baseline, we break down the test results into two subsets: words that were seen in the training dataset (in-vocabulary, or IV), versus words that were not (out-of-vocabulary, or OOV). We removed the entity boundary tags (e.g. “I-ORG” becomes “ORG”) so that each word token is evaluated separately. Dividing IV and OOV subsets can be done by checking each word against a lexicon compiled from training dataset.

If our automatic expert induction LOP method does successfully mitigate the weight undertraining problem, we would expect to see an improvement in OOV recognition. The result of this analysis is shown in Table 3.

On the CoNLL dataset, our method gives significant improvements over MaxEnt in both OOV and IV category. In particular, improvement from



OOV subset (3.4% in  $F_1$ ) is larger than the improvement from IV subset (2% in  $F_1$ ). This matches that our hypothesis that our method mitigates the weight undertraining problem and thus gives a stronger boost in OOV recognition.

This effect is even more pronounced on the MUC dataset. The improvement in IV subset in this case is actually quite modest (0.5%), but we get a significant 2.2% improvement from the OOV subset.

## 7 Related Work

Beyond the several aforementioned works that address the issue of weight undertraining, another recent work that looks at this problem is Wang and Manning (2013). They examined the objective function of dropout training prescribed by Hinton et al. (2012), and proposed an approximation of dropout by directly sampling from a Gaussian approximation. The resulting algorithm is orders of magnitude faster than the iterative algorithm given by Hinton et al. (2012). It will be interesting to compare our proposed method with this method in the future.

The use of group-sparsity norms in NLP research is relatively rare. Martins et al. (2011) proposed the use of structure-inducing norms, in particular  $\ell_1\ell_2$  norm (group lasso), for learning the structure of classifiers. A key observation is that during testing, most of the runtime is consumed in the feature instantiation stage. Since  $\ell_1\ell_2$  norm discards whole groups of features at a time, there are fewer feature templates that need to be instantiated, therefore it could give a significant runtime speedup.

Our use of  $\ell_2\ell_1$  norm takes a different flavor in that we explore the internal structure of the model. There is, however, one recent paper that also makes use of  $\ell_2\ell_1$  norm for a similar reason. Das and Smith (2012) employed  $\ell_2\ell_1$  norm for learning sparse structure in a network formed by lexical predicates and semantic frames. Their results show that  $\ell_2\ell_1$  norm yields much more compact models than  $\ell_1$  or  $\ell_2$  norms, with superior performance in learning to expand lexicons.

However, the optimization problem in Das and Smith (2012) was much simpler since all parameters can only take on positive values, and therefore they could directly use a gradient-descent method with specialized edge condition checks. In our work, we have shown a general-purpose method

in the FOBOS framework for optimizing any convex function with  $\ell_2\ell_1$  norm.

A related  $\ell_1$ - $\ell_2$  mixed norm is called *elastic net* (Zou and Hastie, 2005). It was proposed to overcome a problem of lasso (i.e.,  $\ell_1$  regularization), which occurs when there is a group of correlated predictors, and lasso tends to pick one and ignore all the others. Elastic net takes the form of a sum of a  $\ell_1$  and a  $\ell_2$  norm. It was used in Lavergne et al. (2010) for learning large-scale Conditional Random Fields.

Another popular approach that also explores diversity in model predictions is system combination, which is related to bagging predictors (Breiman, 1996). For example, Sang et al. (2000) reported that by combining systems using different tagging representation and diverse classifier models (e.g., support vector machine, logistic regression, naive Bayes), they were able to achieve significantly higher accuracy than any individual classifier alone. This approach has also been applied extensively in Machine Translation (DeNero et al., 2010) and Speech Recognition (Mikolov et al., 2011).

## 8 Conclusion

Our results show that previous automatic expert definition methods used in logarithmic opinion pools lack robustness across different tasks and data sets. Instead of manually defining good (i.e., diverse) experts, we demonstrated an effective way to induce experts automatically, by using a sparsity-inducing mixed  $\ell_1\ell_2$  norm inspired by elitist lasso. We proposed a novel formulation of the optimization problem with  $\ell_2\ell_1$  norm in the FOBOS framework. Our method gives consistent and significant improvements over a MaxEnt baseline with fine-tuned  $\ell_2$  regularization on two NER datasets. The gains are most evident in recognizing entity types for out-of-vocabulary words.

## Acknowledgments

The authors would like to thank Rob Voigt and the three anonymous reviewers for their valuable comments and suggestions. We gratefully acknowledge the support of the DARPA Broad Operational Language Translation (BOLT) program through IBM. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA or the US government.

## References

- Rie K. Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.
- Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24:123–140.
- Dipanjan Das and Noah A. Smith. 2012. Graph-based lexicon expansion with sparsity-inducing penalties. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. Model combination for machine translation. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934.
- Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.
- Jenny R. Finkel, Trond Grenager, and Christopher D. 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)*.
- Tom Heskes. 1998. Selecting weighting factors in logarithmic opinion pools. In *Proceedings of Advances in Neural Information Processing Systems (NIPS) 10*.
- Geoffery Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.
- Geoffery Hinton. 1999. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN)*.
- Matthieu Kowalski and Bruno Torr sani. 2009. Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients. *Signal, Image and Video Processing*, 3(3):251–264.
- Thomas Lavergne, Olivier Capp , and Fran ois Yvon. 2010. Practical very large scale CRFs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Andr  F. T. Martins, Noah A. Smith, Pedro M. Q. Aguiar, and M rio A. T. Figueiredo. 2011. Structured sparsity in structured prediction. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.
- Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukas Burget, and Jan Cernocky. 2011. Empirical evaluation and combination of advanced language modeling techniques. In *Proceedings 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural language learning (CoNLL)*.
- Erik F. Tjong Kim Sang, Walter Daelemans, Herv  D jean, Rob Koeling, Yuval Krymolowski, Vasin Punyakanok, and Dan Roth. 2000. Applying system combination to base noun phrase identification. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*.
- Andrew Smith, Trevor Cohn, and Miles Osborne. 2005. Logarithmic opinion pools for conditional random fields. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Charles Sutton, Michael Sindelar, and Andrew McCallum. 2007. Reducing weight undertraining in structured discriminative learning. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.
- Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Sida I. Wang and Christopher D. Manning. 2013. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*.
- Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.

# Learning Efficient Information Extraction on Heterogeneous Texts

**Henning Wachsmuth**

University of Paderborn, s-lab  
Paderborn, Germany  
hwachsmuth@s-lab.upb.de

**Benno Stein**

Bauhaus-Universität Weimar  
Weimar, Germany  
benno.stein@uni-weimar.de

**Gregor Engels**

University of Paderborn, s-lab  
Paderborn, Germany  
engels@upb.de

## Abstract

From an efficiency viewpoint, information extraction means to filter the relevant portions of natural language texts as fast as possible. Given an extraction task, different pipelines of algorithms can be devised that provide the same precision and recall but that vary in their run-time due to different pipeline schedules. While recent research investigated how to determine the run-time optimal schedule for a collection or a stream of texts, this paper goes one step beyond: we analyze the run-times of efficient schedules *as a function of the heterogeneity* of the texts and we show how this heterogeneity is characterized from a data perspective. For extraction tasks on heterogeneous big data, we present a self-supervised online adaptation approach that learns to predict the optimal schedule depending on the input text. Our evaluation suggests that the approach will significantly improve efficiency on collections and streams of texts of high heterogeneity.

## 1 Introduction

Information extraction analyzes natural language text in order to find relevant information about entities and the events they participate in. An extraction task often requires to fill event templates with considerable numbers of slots. Such a task implies several analysis steps, e.g. certain types of entity and relation extraction, and it is therefore typically tackled with a pipeline  $\Pi = \langle \mathbf{A}, \pi \rangle$ , where  $\mathbf{A}$  is a set of extraction algorithms and  $\pi$  a schedule that prescribes the order of algorithm application. The information sought for is anchored in text units of a certain size, e.g. in a sentence or paragraph.

In times of big data, the run-time efficiency of information extraction receives much attention in research and industry (Chiticariu et al., 2010).

Among others, a growing need for business intelligence can be regarded as the driving force behind. This trend is equally observed by consulting companies who see evolving markets for predictive analytics (Harper, 2011), by global software players who exploit big data for decision making (White, 2011), and by researchers who seek to annotate tables at web scale (Limaye et al., 2010).

Generally, a pipeline  $\Pi = \langle \mathbf{A}, \pi \rangle$  can be sped up by parallelization (Agichtein, 2005), if given enough resources, or by using faster but less effective algorithms (Al-Rfou' and Skiena, 2012). In addition, information extraction can always be approached as a filtering task as discussed in detail in (Wachsmuth et al., 2013b): By filling a template slot, each algorithm in  $\mathbf{A}$  implicitly classifies certain units of an input text as relevant. Only these units need to be filtered for the next algorithm in  $\pi$ . As a result, a smart schedule  $\pi$  will often significantly improve the overall extraction efficiency. If the input requirements of all algorithms in  $\mathbf{A}$  are met within  $\pi$ , the effectiveness of  $\Pi$  (in terms of both precision and recall) will be maintained, since the output of  $\Pi$  exactly lies in the filtered text units (Wachsmuth and Stein, 2012).<sup>1</sup>

When given a big data filtering task, the designer of a pipeline faces two challenges: (1) How to determine the most efficient schedule for a set of extraction algorithms and a collection or a stream of texts? (2) How to maintain efficiency under heterogeneous text characteristics? With regard to the former challenge we resort to existing research (cf. Section 2). The latter becomes an issue where input texts are not fully known or come from different sources as in the web. Moreover, streams of texts can undergo substantial changes in the distri-

<sup>1</sup>The simplest filtering task is to extract a relation between two entity types, such as  $\langle \text{ORG} \rangle$  was founded in  $\langle \text{TIME} \rangle$ . E.g., the sentence "Google was established by two Stanford students." needs not to be filtered for relation extraction, as it contains no time entity. The schedule of the two implied entity recognition steps will affect the extraction efficiency.

bution of relevant information. We argue that such kinds of uncertainty and lack of a-priori knowledge cannot be tackled offline, but they require to learn and to adapt to the characteristics of input texts to avoid a noticeable efficiency loss.

### 1.1 Contributions and Outline

In this paper, we analyze to what extent the heterogeneity of natural language texts in the distribution of relevant information affects the efficiency of an information extraction pipeline. For a high heterogeneity, we propose an adaptation of the pipeline’s schedule, which we address with online learning. Our learning algorithm maps basic linguistic characteristics of a text to run-times of pipelines and chooses the pipeline with the lowest predicted run-time. The algorithm learns self-supervised and it is language-independent. To measure the impact of heterogeneity, we evaluate the algorithm on precisely constructed text corpora of different heterogeneity. Our contributions are three-fold:

1. We develop a self-supervised online adaptation algorithm that learns the efficiency of information extraction pipelines (Section 3).
2. We quantify the heterogeneity of natural language texts with regard to the distribution of relevant information (Section 4).
3. We evaluate the need for online adaptation in efficient information extraction as a function of the heterogeneity of input texts (Section 5).

## 2 Related Work

One line of research on extraction efficiency refers to *declarative information extraction* (Shen et al., 2007). In particular, Krishnamurthy et al. (2009) created SYSTEMT to address the needs of enterprise extraction applications. SYSTEMT involves optimization strategies such as the ordering and integration of analysis steps (Reiss et al., 2008), but it is restricted to rule-based extraction.

In (Wachsmuth et al., 2011a), we introduced a method that optimizes the schedule of an arbitrary set of extraction algorithms. This method captures much optimization potential and it can be automated using techniques from artificial intelligence (Wachsmuth et al., 2013a). Unlike the approach in this paper, however, the method does not handle variances in the characteristics of input texts.

Our approach applies to all extraction tasks with dependencies between the relevant types of information. We target at template filling, which con-

sists in relating a number of entities to events of predefined types (Cunningham, 2006). Recent research, e.g. (Jean-Louis et al., 2011), and major evaluation tracks, e.g. (Kim et al., 2011), show the ongoing importance of template filling.

We consider extraction pipelines that perform filtering, which have a long tradition (Cowie and Lehnert, 1996). Sarawagi (2008) sees the efficient filtering of relevant portions of input texts as a main challenge. In the pipelines we focus on, each algorithm takes on one analysis (Grishman, 1997). Other approaches such as *joint information extraction* (Choi et al., 2006) can be effective, but they are not suitable when efficiency is important.

van Noord (2009) trades parsing efficiency for parsing effectiveness by learning a heuristic filtering of useful parses. In contrast, we develop a self-supervised online learning algorithm to achieve efficient extraction without reducing effectiveness. While our approach works with every predefined relation and event type, arbitrary binary relations are found in self-supervised *open information extraction* (Fader et al., 2011). Self-supervised learning aims to fully overcome manual text labeling, mostly for learning language like McClosky et al. (2010). To our knowledge, we are the first to apply it for predicting extraction efficiency.

## 3 Learning Efficient Extraction

The run-time efficiency of an information extraction pipeline depends on the distribution of relevant information in its input texts (Wachsmuth and Stein, 2012). For situations where this distribution varies, we now present an approach that chooses a pipeline schedule depending on the text at hand. To maintain precision and recall, we consider only schedules that fulfill the input requirements of all algorithms employed in a pipeline.

### 3.1 Splitting the Pipeline

Most extraction tasks require some analyses (e.g. tokenization) to be performed on the whole input texts, as they are needed for most or all subsequent analyses. We exploit this notion in that we use the results of the first algorithms in a pipeline to predict the best schedule of the remaining algorithms. To this end, we split a pipeline into a fixed first part and a variable second part. We call the first part the *prefix pipeline*, denoted as  $\Pi_{pre}$ , and each second part  $\Pi_1, \dots, \Pi_k$  a *main pipeline*.

In general,  $k$  has an upper bound of  $m!$  where  $m$  is the number of algorithms in a main pipeline.

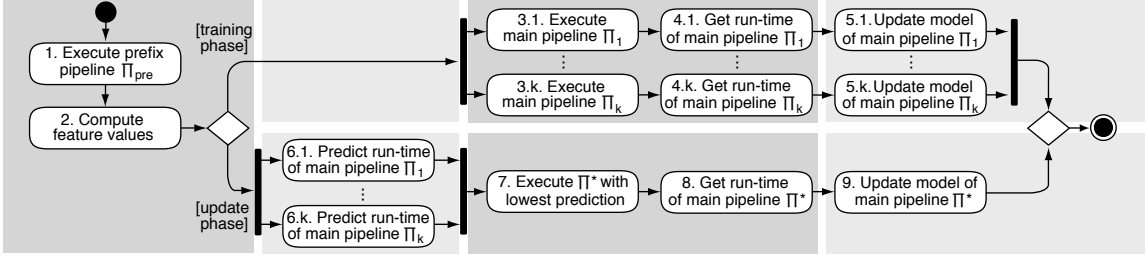


Figure 1: Illustration of the online adaptation algorithm with a prefix pipeline  $\Pi_{pre}$  and  $k$  main pipelines  $\Pi_1, \dots, \Pi_k$  for one input text, which either goes through the training phase or through the update phase.

Due to the algorithms' input constraints, however,  $k$  is normally much lower in practice. Also, there might be ways to restrict the set of candidate schedules to a reasonable selection, which is itself a non-trivial problem that is beyond the scope of this paper. In the following, we simply assume that  $k \leq m!$  main pipelines are given.

### 3.2 Self-Supervised Learning of Run-times

For a collection or a stream of texts  $\mathbf{D}$ , our goal is to determine the most efficient main pipeline for each text in  $\mathbf{D}$ . We approach this goal as an online regression problem by learning to predict the run-time per text unit  $t(\Pi_i)$  of each main pipeline  $\Pi_i \in \{\Pi_1, \dots, \Pi_k\}$ . Based on the results of  $\Pi_{pre}$ , we represent each text  $D \in \mathbf{D}$  as a feature vector  $(x_1, \dots, x_p)$  in order to create a regression model for each  $t(\Pi_i)$ . Concretely, we map the feature values  $x_1^{(D)}, \dots, x_p^{(D)}$  for  $D$  to a predicted run-time  $\tilde{t}(\Pi_i)$ . Then,  $D$  is processed by the main pipeline  $\Pi^*$  with the lowest prediction.

In this manner, learning can be approached self-supervised, as all training data is generated automatically: the feature values and the observed run-time  $t(\Pi^*)$  of  $\Pi^*$  on  $D$  serve as a new training instance, and the prediction error is given by the difference between  $\tilde{t}(\Pi^*)$  and  $t(\Pi^*)$ . Still, an explicit training set that is processed by all main pipelines helps to create initial regression models.

### 3.3 The Online Adaptation Algorithm

Let a prefix pipeline  $\Pi_{pre}$ , a set of main pipelines  $\Pi_1, \dots, \Pi_k$ , and a collection or a stream of texts  $\mathbf{D}$  be given. Then  $\mathbf{D}$  is split into two parts  $\mathbf{D}_T$  and  $\mathbf{D}_U$  to serve the following two phases of the *online adaptation algorithm*:

1. *Training*. On each text  $D \in \mathbf{D}_T$ , execute  $\Pi_{pre}$  and each  $\Pi_i \in \{\Pi_1, \dots, \Pi_k\}$ . Update the regression model of each  $\Pi_i$  wrt. the results of  $\Pi_{pre}$  and the run-time  $t(\Pi_i)$  of  $\Pi_i$  on  $D$ .

2. *Update*. On each text  $D \in \mathbf{D}_U$ , execute  $\Pi_{pre}$  and predict  $\tilde{t}(\Pi_i)$  for all  $\Pi_i \in \{\Pi_1, \dots, \Pi_k\}$ . Execute the  $\Pi^*$  with the lowest prediction and update its regression model wrt. the results of  $\Pi_{pre}$  and the run-time  $t(\Pi^*)$  of  $\Pi^*$  on  $D$ .

Figure 1 illustrates the online adaptation algorithm on one single text. An intuitive extension is to iteratively schedule each extraction algorithm separately. This would allow us to use detailed knowledge in later predictions. Since the first predictions are most decisive, however, we do not consider the iterative scenario here for simplicity.

### 3.4 Baselines and the Gold Standard

One way to evaluate our approach is to compare it to each pipeline  $(\Pi_{pre}, \Pi_i)$ ,  $\Pi_i \in \{\Pi_1, \dots, \Pi_k\}$ . In practice, relying on such a fixed pipeline involves the danger of choosing a slow one. Hence, we also consider two baseline approaches below:

1. *Random baseline*. For each text  $D \in \mathbf{D}_U$ , choose a main pipeline (pseudo-) randomly.
2. *Optimal baseline*. For each text  $D \in \mathbf{D}_U$ , choose the main pipeline that has achieved the best overall run-time on  $\mathbf{D}_T$ .

The optimal baseline serves as a strong competitor on homogeneous collections and streams of texts, where it will often find the run-time optimal fixed pipeline. In contrast, the random baseline appears rather weak, but it will never fail completely.

Besides, we compute the *gold standard* below, i.e., an oracle that knows the fastest main pipeline for each text beforehand. The gold standard defines the upper ceiling for a set of main pipelines, thereby quantifying the general optimization potential. In that, it helps to evaluate whether online adaptation is suitable for the input texts at hand.

## 4 The Heterogeneity of Texts

The introduced algorithm is domain-independent and language-independent. It targets at situations

where input texts are heterogeneous in content or style, as is typical for the results of an exploratory web search. From an extraction perspective, the heterogeneity of a collection or a stream of texts  $\mathbf{D}$  can be regarded as the extent to which the texts in  $\mathbf{D}$  vary in the distribution of instances of the relevant *information types*  $C_1, \dots, C_m$ , i.e., the entity, relation, and event types to be extracted.

As motivated in Section 1, text units need to be analyzed only if they may contain instances of all relevant information types. Hence, a pipeline’s efficiency depends on the *density*  $\rho_i(\mathbf{D})$  of each type  $C_i$  in the input texts in  $\mathbf{D}$ . Here, the density corresponds to the fraction of text units in  $\mathbf{D}$  that contain an instance of  $C_i$ . The density  $\rho_i(D)$  of  $C_i$  in a single text  $D \in \mathbf{D}$  can be defined accordingly. Now, differences in the run-time per text unit of a pipeline mainly result from varying densities  $\rho_i(D)$ . In this regard, the heterogeneity of  $\mathbf{D}$  can be quantified by measuring the variance of all densities in the texts in  $\mathbf{D}$ . The outlined considerations give rise to the following measure:

**Averaged Deviation** Let  $\mathbf{C} = \{C_1, \dots, C_m\}$  be the set of relevant information types for an extraction task, and let  $\sigma_i(\mathbf{D})$  be the standard deviation of the density  $\rho_i(\mathbf{D})$  of  $C_i \in \mathbf{C}$  in a collection or a stream of texts  $\mathbf{D}$ . Then, the averaged deviation of  $\mathbf{C}$  in  $\mathbf{D}$  is

$$\sigma(\mathbf{C}|\mathbf{D}) = \frac{1}{m} \cdot \sum_{i=1}^m \sigma_i(\mathbf{D}).$$

We compute exact values  $\sigma(\mathbf{C}|\mathbf{D})$  in Section 5 to measure the impact of heterogeneity. In general, the averaged deviation can also be estimated on a sample of texts. For illustration, Table 1 lists the deviations for the three most common named entity types in the German part of the *CoNLL’03 corpus* (Tjong Kim Sang and De Meulder, 2003), in the *Revenue corpus* (Wachsmuth et al., 2010), in a sample of the German Wikipedia (the first 10,000 articles according to internal page ID), and in the *LFA-11 smartphone corpus*, which is a web crawl of blog posts (Wachsmuth and Bujna, 2011). Here, we recognized entities using *Stanford NER* (Finkel et al., 2005; Faruqui and Padó, 2010).

Different from other sampling-based efficiency estimations, cf. (Wang et al., 2011), the averaged deviation does *not* measure the typical characteristics of input texts, but it quantifies how much these characteristics vary. By that, it helps pipeline designers to decide whether an online adaptation of pipeline schedules is needed to ensure efficient ex-

Information type $C_i$	$\sigma_i(\mathbf{D}_{\text{co}})$	$\sigma_i(\mathbf{D}_{\text{rv}})$	$\sigma_i(\mathbf{D}_{\text{wk}})$	$\sigma_i(\mathbf{D}_{\text{bp}})$
Person entities	18.4%	11.1%	15.9%	16.6%
Organization entities	18.1%	16.0%	14.1%	23.4%
Location entities	16.6%	10.9%	16.0%	15.3%
<b>Averaged deviation</b>	<b>17.7%</b>	<b>12.7%</b>	<b>15.3%</b>	<b>18.4%</b>

Table 1: The standard deviation  $\sigma_i$  of the density of three entity types in the CoNLL’03 corpus  $\mathbf{D}_{\text{co}}$ , the Revenue corpus  $\mathbf{D}_{\text{rv}}$ , a sample of 10,000 Wikipedia articles  $\mathbf{D}_{\text{wk}}$ , and a crawl of blog posts  $\mathbf{D}_{\text{bp}}$ . The bottom line shows their averaged deviations.

traction. However, its current form leaves unclear how to compare deviations across tasks. In future work, a solution will be to normalize the averaged deviation—either with respect to a reference corpus or with respect to the given task, e.g. to a situation where all schedules perform equally well.

#### 4.1 Text Corpora of Different Heterogeneity

For a careful evaluation of online adaptation, we need input texts that refer to different levels of heterogeneity while being appropriate for analyzing a single and sufficiently complex filtering task at the same time. Most corpora for extraction tasks are too small to create reasonable subsets of different heterogeneity like  $\mathbf{D}_{\text{co}}$  and  $\mathbf{D}_{\text{rv}}$ . As an alternative, a web crawl such as  $\mathbf{D}_{\text{bp}}$  typically yields high heterogeneity, but it tends to include a large fraction of task-irrelevant texts. This conceals which efficiency differences are due to scheduling and, thus, is not suitable for controlled experiments.

To address this difficulty, we also use precisely constructed text corpora below, which consist of both original texts from existing corpora and artificially modified versions of these texts. Concretely, we modified a text by randomly duplicating one of its sentences, ensuring that each text in a corpus comprises a unique set of sentences while being grammatically valid. Thereby, we limit the online adaptation algorithm to a certain degree in learning linguistic features from the texts, but we gain that we can measure the benefit of online adaptation as a function of the averaged deviation.

## 5 Evaluation

We now present controlled experiments with the online adaptation algorithm on text corpora of different heterogeneity. The goal is to show the circumstances under which online adaptation will be needed for efficiency and, conversely, when a run-time optimal fixed pipeline appears sufficient.

## 5.1 Experimental Set-up

We consider the filtering task to extract financial forecast statements with resolvable time information and a monetary value for an organization. An example for such a statement is “*Apple’s annual revenues could hit \$400 billion by 2015*”. Accordingly, we have a set  $C$  of five relevant information types: time entities, money entities, organization entities, financial statements, and forecast events.

**Algorithms** Table 2 outlines the eight algorithms that we used in all experiments. We employed the *UIMA tokenizer*<sup>2</sup> to generate tokens and sentences, and the *TreeTagger* for part-of-speech tagging and chunking (Schmid, 1995). As in (Wachsmuth et al., 2011a), we relied on regexes for money and time recognition, while we applied support vector machines SD and FD for event detection. Organization names were extracted with the CRF-based *Stanford NER* (cf. Section 4). Finally, we implemented a rule-based time normalizer TN.

While, in general, our approach works for each kind of extraction algorithm, all algorithms in Table 2 perform only in-sentence extraction.

**Pipelines** As stated in Section 3, we split all pipelines into two parts. The prefix pipeline consists of the two algorithms used for preprocessing only:  $\Pi_{pre} = (UT, TT)$ . For the reasons given below, we evaluated the following  $k = 3$  main pipelines:

$$\Pi_1 = (TR, FD, MR, SD, TN, OR)$$

$$\Pi_2 = (TR, MR, FD, TN, OR, SD)$$

$$\Pi_3 = (MR, TR, FD, OR, SD, TN)$$

Only 108 of the  $6! = 720$  possible main pipelines fulfill all dependencies listed in Table 2. Based on the method from (Wachsmuth et al., 2011a), we found that main pipelines starting with TR, MR, and FD (which are significantly faster than OR, SD, and TN) dominate others. We selected  $\Pi_1$ ,  $\Pi_2$ , and  $\Pi_3$ , as they target at very different distributions of relevant information. While  $k = 3$  appears small, it allows for a concise evaluation and suffices to discuss the impact of online adaptation. Still, we evaluate all 108 main pipelines in Section 5.4.

**Features** For generality, we restricted our view to simple features that neither require a preceding run over the training set nor exploit knowledge about the employed algorithms: (1) *Lexical statistics*, namely, the average and maximum number of characters in a token and of tokens in a sentence as

<sup>2</sup>UIMA tokenizer, <http://uima.apache.org/sandbox.html>.

Algorithm $A$	$t(A)$	depends on
UT UIMA tokenizer	0.06 ms	–
TT TreeTagger	0.59 ms	UT
TR Time recognition	0.36 ms	UT
MR Money recognition	0.64 ms	UT
OR Organization recognition	2.52 ms	UT, TT
SD Financial statement detection	3.95 ms	UT, TR, MR
FD Forecast event detection	0.29 ms	UT, TT, TR
TN Time normalization	0.95 ms	UT, TR

Table 2: Each evaluated algorithm  $A$  with its estimated average run-time per sentence  $t(A)$  and the algorithm  $A$  depends on.

well as the length of the text, (2) the average *run-times* per sentence of each algorithm in  $\Pi_{pre}$ , and (3) the frequencies of all *part-of-speech tags*.

In the feature evaluation in Section 5.4, we also have two further types that capture general characteristics of entities: (4) The frequency of each unigram and bigram of all *chunk tags* and (5) the frequencies of *regex matches* of a regex for arbitrary numbers and of a regex for upper-case words.<sup>3</sup>

**Learning Algorithm** For run-time prediction, we applied *Stochastic Gradient Descent (SGD)* from *Weka 3.7.5* (Hall et al., 2009). After some preliminary tests, we set the learning rate of SGD to 0.01 for all experiments. Accordingly, we always used  $10^{-5}$  for regularization and we always let SGD iterate 10 epochs over the training texts.

**Datasets** We constructed four partly artificial corpora  $D_0, \dots, D_3$  as motivated in Section 4.1. In case of  $D_0$ , we randomly mixed 1500 texts of the German CoNLL’03 corpus and the Revenue corpus (cf. Section 4).<sup>4</sup> For  $D_1$ , we took the 300 texts from  $D_0$  with the highest differences in the density of relevant information. We created modified versions of these texts in order to obtain a corpus size of 1500.  $D_2$  and  $D_3$  were built analogously for the 200 and 100 highest-difference texts, respectively. Table 3 lists all averaged deviations for the text unit “sentence”. Where not stated otherwise, we used the first 500 texts of each corpus for training and the remaining 1000 for testing.

**Efficiency** The efficiency of all pipelines on each text was measured as the *run-time in milliseconds per sentence*, averaged over 10 runs. All run-times and their standard deviations were saved. For determinism, we loaded these run-times during eval-

<sup>3</sup>We experimented with further regexes, but their impact was low. Therefore, we do not report on them in this paper.

<sup>4</sup>Notice that the evaluated set of features does not target at characteristics that are specific to the German language.

Information type	$C_i$	$\sigma_i(\mathbf{D}_0)$	$\sigma_i(\mathbf{D}_1)$	$\sigma_i(\mathbf{D}_2)$	$\sigma_i(\mathbf{D}_3)$
Time entities		19.1%	22.5%	24.6%	25.9%
Money entities		19.8%	19.1%	20.4%	22.3%
Organization entities		19.3%	21.6%	22.4%	25.0%
Financial statements		7.1%	7.8%	8.9%	10.6%
Forecast events		3.8%	5.9%	6.7%	8.5%
<b>Averaged deviation</b>		<b>13.8%</b>	<b>15.4%</b>	<b>16.6%</b>	<b>18.5%</b>

Table 3: The standard deviation  $\sigma_i(\mathbf{D})$  of the density of each relevant information types  $C_i \in \mathbf{C}$  for each corpus  $\mathbf{D} \in \{\mathbf{D}_0, \dots, \mathbf{D}_3\}$  as well as the averaged deviation  $\sigma(\mathbf{C}|\mathbf{D})$  of  $\mathbf{C}$  in each  $\mathbf{D}$ .

uation instead of executing pipelines.<sup>5</sup> In case of the online adaptation algorithm, we also computed the *mean run-time prediction error* and the *classification error*, i.e., the fraction of texts the best main pipeline was not found for.

**Effectiveness** We omit to evaluate effectiveness here for lack of relevance: Our experiment setting, which is similar to (Wachsmuth et al., 2011a), yields no trade-off between efficiency and effectiveness, since we only consider schedules that fulfill all dependencies in Table 2. Thus, all pipelines always achieve the same precision and recall.

**System and Software** All experiments were conducted on a 2 GHz Intel Core 2 Duo MacBook with 4 GB memory. The Java source code and the pre-computed run-time files used in the evaluation can be accessed at <http://www.arguana.com>.

## 5.2 The Impact of Heterogeneity

We ran the online adaptation algorithm and the baselines from Section 3 on the test set of each of the corpora  $\mathbf{D}_0, \dots, \mathbf{D}_3$ . Both the algorithm and the optimal baseline were trained on the respective training sets. Figure 2 illustrates the run-times of the main pipelines for each approach as a function of the averaged deviation and compares them to the gold standard. The shown confidence intervals result from the run-times’ standard deviations  $\sigma$ , which ranged between 0.029ms and 0.043ms.

On the least heterogeneous corpus  $\mathbf{D}_0$  with an averaged deviation of 13.8%, the online adaptation algorithm achieved an average run-time of 0.98ms per sentence, which is faster than the random baseline but slower than the optimal baseline at a low confidence level. For  $\sigma(\mathbf{C}|\mathbf{D}_1) = 15.4\%$ , the online adaptation algorithm succeeded with 0.87ms per sentence as opposed to 0.9ms of the optimal baseline. This gap gets significantly larger un-

<sup>5</sup>Section 5.4 shows the effects of errors of measurement.

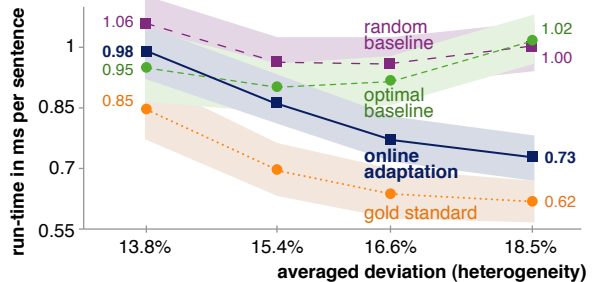


Figure 2: The average run-times of the main pipelines of both baseline approaches, the online adaptation algorithm, and the gold standard as a function of the averaged deviation. The background areas denote the 95% confidence intervals ( $\pm 2\sigma$ ).

der higher averaged deviation. At 18.5%, both baselines are clearly outperformed, taking 37% and 40% more time on average, respectively.<sup>6</sup>

One reason for the weak result of online adaptation on  $\mathbf{D}_0$  lies in the low optimization potential for that corpus: the main pipeline of the optimal baseline took only 12% more time on average than the gold standard (0.95ms vs. 0.85ms), which implies very small differences in the main pipelines’ run-times. This does not only render online adaptation hard but also unnecessary. Conversely, Figure 3 shows an optimization potential of over 50% for  $\mathbf{D}_3$ . A reasonable hypothesis is therefore that online adaptation succeeds only on collections and streams of texts of high heterogeneity as indicated by large differences in the pipelines’ run-times.

## 5.3 Run-time and Error Analysis

Figure 3 details the run-times of the three fixed pipelines, the online adaptation algorithm, and the gold standard on  $\mathbf{D}_0$  and  $\mathbf{D}_3$ . The small black indicators denote the standard deviations.

In total, the fixed pipelines are 16% to 25% slower than the online adaptation algorithm on  $\mathbf{D}_3$ . The algorithm’s run-time mainly breaks down into 0.51ms of the prefix pipeline and 0.73ms of the main pipelines, while the time for feature computations (0.03ms) and regression (0.01ms) is almost negligible. A similar situation is observed for  $\mathbf{D}_0$ . Here, online adaptation is worse only than  $(\Pi_{pre}, \Pi_1)$ , which did best on 598 of the 1000 test texts.  $(\Pi_{pre}, \Pi_2)$  and  $(\Pi_{pre}, \Pi_3)$  had the lowest run-time on 229 and 216 texts, respectively.<sup>7</sup>

<sup>6</sup>On the training set of  $\mathbf{D}_3$ , the optimal baseline did not find the fastest main pipeline. This might be coincidence, but is, of course, more likely under higher heterogeneity.

<sup>7</sup>The numbers of texts sum up to more than 1000 because on some texts different pipelines performed equally well.



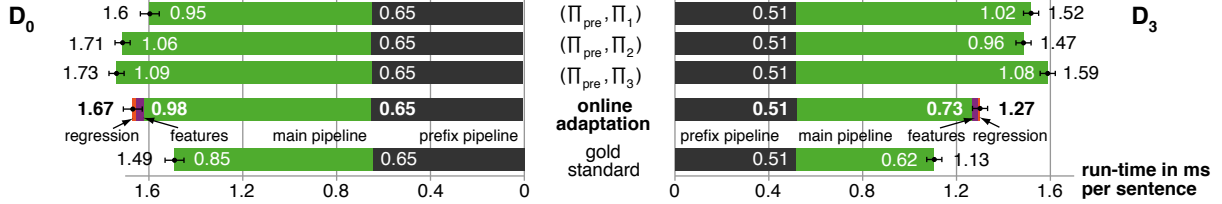


Figure 3: Average run-times of the fixed pipelines and online adaptation on the test sets of  $D_0$  and  $D_3$ .

The online adaptation algorithm did not succeed on  $D_0$ , since its mean run-time prediction error of 0.45ms was almost half as high as the average run-times to be predicted, which is too inaccurate under the small differences in the pipelines’ run-times. As a result, for only 39% of the input texts, the best pipeline was chosen (i.e., a classification error of 0.61). However, the impact on  $D_3$  does not emanate from a low mean prediction error (that was in fact 0.24ms higher), but the classification error was reduced to 0.45. Consequently, the main reason lies in larger differences in the pipelines’ run-times, which supports our hypothesis.

An insightful linguistic phenomenon is that the prefix pipeline  $\Pi_{pre}$  took significantly more time per sentence on  $D_0$  than on  $D_3$ . Since the run-times of both algorithms in  $\Pi_{pre}$  scale linearly with the number of input tokens, the average sentence length of  $D_0$  must exceed that of  $D_3$ . The reason is that shorter sentences tend to contain less relevant information. Hence, many sentences can be discarded after a few analysis steps, which increases the need for input-dependent scheduling.

#### 5.4 Parameter Analysis

To give further evidence for the hypothesis from Section 5.2 and to test the applicability of online adaptation, we evaluated some major parameters:

**Impact of Features** For each of the five feature types in isolation, we trained a regression model on  $D_0$  and analyzed its impact. The *lexical statistics* achieved the lowest classification error (0.41), followed by the *run-times* (0.53). In terms of run-time prediction, the *regex matches* (0.46ms) and the *part-of-speech tags* (0.48ms) performed best, whereas the *chunk tags* failed both in classification (0.57) and prediction (0.58ms).<sup>8</sup> We used feature types 1–3 in all other experiments, since comput-

<sup>8</sup>The low correlation of the prediction and classification error seems counterintuitive, but it indicates the limitations of these measures: E.g, a small prediction error can still be problematic if run-times differ only slightly, while a high classification error may have few negative effects in this case. If both errors are small, however, this normally implies success.

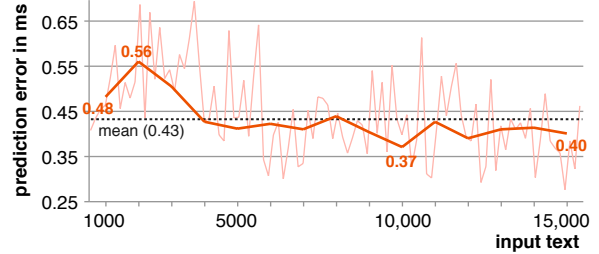


Figure 4: The mean run-time prediction error for the pipelines chosen by the online adaptation algorithm on 15,000 modified versions of the texts in  $D_0$  for training size 1. The values of the two interpolated learning curves denote the mean of 1000 and 100 consecutive predictions, respectively.

ing them took only 0.05ms per sentence on average. In contrast, the *regex matches* needed 0.16ms alone, which exceeds the difference between the optimal baseline and the gold standard on  $D_0$  (cf. Section 5.2) and, thus, renders the *regex matches* useless in the given setting.

The *regex matches* emphasize the obvious need for a scheduling mechanism that avoids spending more time than can be saved later on. At the same time, such a mechanism should capture characteristics of a text that reliably model its complexity, which the evaluated features did not fully achieve.

**Impact of Training Size** We evaluated the online adaptation algorithm on  $D_0$  for nine training sizes between 1 and 5000. As the training set of  $D_0$  is limited, we created modified versions of its texts where needed (cf. Section 4.1). Online adaptation always did better than the random baseline but not than the optimal baseline except for training size 1. In case of 1000 or more training texts, the algorithm mimicked the optimal baseline, i.e., it chose  $\Pi_1$  for about 90% of the texts.

**Online Learning** For training size 1, we ran the online adaptation algorithm on 15,000 modified versions of the texts in  $D_0$ . Figure 4 shows two levels of detail of the algorithm’s learning curve in its update phase. As the bold curve conveys, the mean run-time prediction error decreases on the

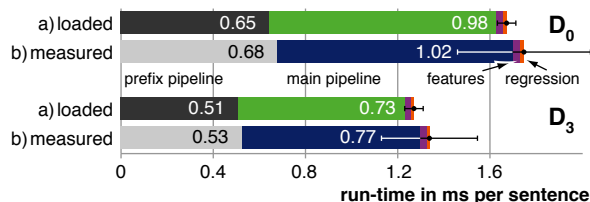


Figure 5: The average run-times and standard deviations of the online adaptation algorithm on  $D_0$  and  $D_3$  when run-times are a) loaded from a pre-computed file or b) measured during execution.

first 5000 texts to an area around 0.4ms where it stays most of the time afterwards, though the light curve discovers many outliers. Still, online learning apparently seems to work well.

**Overall Optimization Potential** To measure the overall optimization potential of scheduling, we made an experiment with all  $k = 108$  correct main pipelines on  $D_0$  (cf. Section 5.1).<sup>9</sup> The resulting average run-times of the fixed main pipelines span from 0.79ms per sentence in case of  $\Pi_{best} = (TR, FD, MR, OR, TN, SD)$  to 3.27ms of  $\Pi_{worst} = (OR, TR, TN, MR, SD, FD)$ . This shows that the efficiency loss of choosing a wrong schedule can be very high. The online adaptation algorithm achieved 0.86ms with a mean run-time prediction error of 0.37ms. Altogether, 21 of the 108 schedules were used on the 1000 test texts, and the best schedule was chosen for 30% of the texts.

**Real Execution** As mentioned, we loaded all run-times from a precomputed file, which is not possible in case of real execution. In Figure 5, we compare the results of loading run-times to the run-times measured during the execution of the online adaptation algorithm. The measured values mainly differ in terms of larger standard deviations, i.e., 0.16ms on  $D_0$  and 0.12ms on  $D_3$ . This seems to have a fairly negative effect on the main pipelines’ run-times. However, the measured prefix pipeline run-times also exceed the saved ones, which suggests that the effect is due to a higher system load only. In any case, the measured run-time on  $D_3$  indicates that the online adaptation algorithm applies for practical applications.

## Conclusion

In this paper, we analyze the efficiency of information extraction pipelines as a function of the het-

<sup>9</sup>Notice that the training time increases linear to  $k$ , so a high  $k$  implies a high training overhead. For space reasons, we omit to report on training time here at all. However, training time will always be amortized in large-scale scenarios.

erogeneity of their input texts. In particular, we quantify heterogeneity with regard to the distribution of relevant information and we provide a self-supervised online adaptation algorithm that learns which pipeline schedule to choose for what input text in order to optimize efficiency while maintaining precision and recall. On this basis, we investigate the need for pipelines that adapt to their input within time-critical extraction tasks.

Our experiments suggest that the benefit of online adaptation is significant on heterogeneous collections and streams of texts: The online adaptation algorithm achieves gains of about 30% over the most efficient fixed schedule, which we see as important in times of big data. Conversely, when relevant information is uniformly distributed, finding an efficient fixed schedule appears sufficient, as approached in (Wachsmuth et al., 2013a).

A setting still to be evaluated refers to streams of input texts whose characteristics change slowly over time. Also, other extraction tasks may yield more insights. In order to decide how to approach a task at hand, a better understanding of the processing complexity of collections and streams of texts is required, to which our research contributes substantial building blocks.

In general, our self-supervised learning concept can be transferred to each natural language processing task that meets two basic conditions: (1) The task can be approached in different manners where each approach performs best for certain situations or inputs. (2) The performance of each approach can be measured or it is clear by definition.

## Acknowledgments

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) under contract number 01IS11016A.

## References

- Eugene Agichtein. 2005. Scaling Information Extraction to Large Document Collections. *Bulletin of the IEEE Computer Society TCDE*, 28:3–10.
- Rami Al-Rfou’ and Steven Skiena. 2012. SpeedRead: A Fast Named Entity Recognition Pipeline. In *Proc. of the 24th COLING*, pages 51–66.
- Laura Chiticariu, Yunyao Li, Sriram Raghavan, and Frederick R. Reiss. 2010. Enterprise Information Extraction: Recent Developments and Open Challenges. In *Proc. of the 16th COMAD*, pages 1257–1258.

- Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint Extraction of Entities and Relations for Opinion Recognition. In *Proc. of the 2006 EMNLP*, pages 431–439.
- Hamish Cunningham. 2006. Information Extraction, Automatic. *Encyclopedia of Language & Linguistics*, 4:665–677.
- Jim Cowie and Wendy Lehnert. 1996. Information Extraction. *Communications of the ACM*, 39(1):80–91.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proc. of the 2011 EMNLP*, pages 1535–1545.
- Jenny R. Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proc. of the 43rd Annual Meeting of the ACL*, pages 363–370.
- Manaal Faruqui and Sebastian Padó. 2010. Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In *Proc. of KONVENS 2010*, pages 129–133.
- Ralph Grishman. 1997. Information Extraction: Techniques and Challenges, In *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, pages 10–27.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18.
- Fern Harper. 2011. *Predictive Analytics: The Hurwitz Victory Index Report*, Hurwitz & Associates.
- Ludovic Jean-Louis, Romaric Besançon, and Olivier Ferret. Text Segmentation and Graph-based Method for Template Filling in Information Extraction. In *Proc. of the 5th IJCNLP*, pages 723–731, 2011.
- Jin-D. Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011. Overview of BioNLP Shared Task 2011. In *Proc. of the BioNLP 2011 Workshop Companion Volume for Shared Task*, pages 1–6.
- Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick R. Reiss, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2009. SystemT: A System for Declarative Information Extraction. In *SIGMOD Records*, 37(4):7–13.
- Girija Limaye, Sunita Sarawagi, Soumen Chakrabarti. 2010. Annotating and Searching Web Tables using Entities, Types and Relationships. In *Proc. of the VLDB Endowment*, 3(1):1338–1347.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, USA.
- David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic Domain Adaptation for Parsing. In *Proc. of the HLT/NAACL*, pages 28–36.
- Gertjan van Noord. 2009. Learning Efficient Parsing. In *Proc. of the 12th EACL*, pages 817–825.
- Frederick R. Reiss, Sriram Raghavan, Rajasekar Krishnamurthy, Huaiyu Zhu, and Shivakumar Vaithyanathan. 2008. An Algebraic Approach to Rule-Based Information Extraction. In *Proc. of the 2008 IEEE 24th ICDE*, pages 933–942.
- Sunita Sarawagi. 2008. Information Extraction. *Foundations and Trends in Databases*, 1(3):261–377.
- Helmut Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the ACL SIGDAT-Workshop*, pages 47–50.
- Warren Shen, AnHai Doan, Jeffrey F. Naughton, and Raghu Ramakrishnan. 2007. Declarative Information Extraction using Datalog with Embedded Extraction Predicates. In *Proc. of the 33rd VLDB*, pages 1033–1044.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition In *Proc. of the 7th CoNLL*, pages 142–147.
- Henning Wachsmuth, Peter Prettenhofer, and Benno Stein. 2010. Efficient Statement Identification for Automatic Market Forecasting. In *Proc. of the 23rd COLING*, pages 1128–1136.
- Henning Wachsmuth, Benno Stein, and Gregor Engels. 2011. Constructing Efficient Information Extraction Pipelines. In *Proc. of the 20th ACM CIKM*, pages 2237–2240.
- Henning Wachsmuth and Kathrin Bujna. 2011. Back to the Roots of Genres: Text Classification by Language Function. In *Proc. of the 5th IJCNLP*, pages 632–640.
- Henning Wachsmuth and Benno Stein. 2012. Optimal Scheduling of Information Extraction Algorithms. In *Proc. of the 24th COLING: Posters*, pages 1281–1290.
- Henning Wachsmuth, Mirko Rose, and Gregor Engels. 2013. Automatic Pipeline Construction for Real-Time Annotation. In *Proc. of the 14th CICLing*, pages 38–49.
- Henning Wachsmuth, Benno Stein, and Gregor Engels. 2013. Information Extraction as a Filtering Task. To appear in *Proc. of the 22th ACM CIKM*.
- Daisy Z. Wang, Long Wei, Yunyao Li, Frederick R. Reiss, and Shivakumar Vaithyanathan. 2011. Selectivity Estimation for Extraction Operators over Text Data. In *Proc. of the 2011 IEEE 27th ICDE*, pages 685–696.
- Colin White. 2011. *Using Big Data for Smarter Decision Making*, BI Research.

# TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction

Adrien Bougouin and Florian Boudin and Béatrice Daille

Université de Nantes, LINA, France

{adrien.bougouin,florian.boudin,beatrice.daille}@univ-nantes.fr

## Abstract

Keyphrase extraction is the task of identifying single or multi-word expressions that represent the main topics of a document. In this paper we present TopicRank, a graph-based keyphrase extraction method that relies on a topical representation of the document. Candidate keyphrases are clustered into topics and used as vertices in a complete graph. A graph-based ranking model is applied to assign a significance score to each topic. Keyphrases are then generated by selecting a candidate from each of the top-ranked topics. We conducted experiments on four evaluation datasets of different languages and domains. Results show that TopicRank significantly outperforms state-of-the-art methods on three datasets.

## 1 Introduction

Keyphrases are single or multi-word expressions that represent the main topics of a document. Keyphrases are useful in many tasks such as information retrieval (Medelyan and Witten, 2008), document summarization (Litvak and Last, 2008) or document clustering (Han et al., 2007). Although scientific articles usually provide them, most of the documents have no associated keyphrases. Therefore, the problem of automatically assigning keyphrases to documents is an active field of research.

Automatic keyphrase extraction methods are divided into two categories: supervised and unsupervised methods. Supervised methods recast keyphrase extraction as a binary classification task (Witten et al., 1999), whereas unsupervised methods apply different kinds of techniques such as language modeling (Tomokiyo and Hurst, 2003), clustering (Liu et al., 2009) or graph-based ranking (Mihalcea and Tarau, 2004).

In this paper, we present a new unsupervised method called TopicRank. This new method is an improvement of the TextRank method applied to keyphrase extraction (Mihalcea and Tarau, 2004). In the TextRank method, a document is represented by a graph where words are vertices and edges represent co-occurrence relations. A graph-based ranking model derived from PageRank (Brin and Page, 1998) is then used to assign a significance score to each word. Here, we propose to represent a document as a complete graph where vertices are not words but topics. We define a topic as a cluster of similar single and multi-word expressions.

Our approach has several advantages over TextRank. Intuitively, ranking topics instead of words is a more straightforward way to identify the set of keyphrases that covers the main topics of a document. To do so, we simply select a keyphrase candidate from each of the top-ranked clusters. Clustering keyphrase candidates into topics also eliminates redundancy while reinforcing edges. This is very important because the ranking performance strongly depends on the conciseness of the graph, as well as its ability to precisely represent semantic relations within a document. Hence, another advantage of our approach is the use of a complete graph that better captures the semantic relations between topics.

To evaluate TopicRank, we follow Hasan and Ng (2010) who stated that multiple datasets must be used to evaluate and fully understand the strengths and weaknesses of a method. We use four evaluation datasets of different languages, document sizes and domains, and compare the keyphrases extracted by TopicRank against three baselines (TF-IDF and two graph-based methods). TopicRank outperforms the baselines on three of the datasets. As for the fourth one, an additional experiment shows that an improvement could be achieved with a more effective selection strategy.

The rest of this paper is organized as follows. Section 2 presents the existing methods for the keyphrase extraction task, Section 3 details our proposed approach, Section 4 describes the evaluation process and Section 5 shows the analyzed results. Finally, Section 6 concludes this work and suggests directions for future work.

## 2 Related Work

The task of automatic keyphrase extraction has been well studied and many supervised and unsupervised approaches have been proposed. For supervised methods, keyphrase extraction is often treated as a binary classification task (Witten et al., 1999). Unsupervised approaches proposed so far have involved a number of techniques, including language modeling (Tomokiyo and Hurst, 2003), clustering (Liu et al., 2009) and graph-based ranking (Mihalcea and Tarau, 2004). While supervised approaches have generally proven to be more successful, the need for training data and the bias towards the domain on which they are trained remain two critical issues.

In this paper, we concentrate on graph-based ranking methods for keyphrase extraction. Starting with TextRank (Mihalcea and Tarau, 2004), these methods are becoming the most widely used unsupervised approaches for keyphrase extraction. In TextRank, a document is represented as a graph in which vertices are words connected if they co-occur in a given window of words. The significance of each vertex is computed using a random walk algorithm derived from PageRank (Brin and Page, 1998). Words corresponding to the top ranked vertices are then selected and assembled to generate keyphrases.

Wan and Xiao (2008) propose SingleRank, a simple modification of TextRank that weights the edges with the number of co-occurrences and no longer extracts keyphrases by assembling ranked words. Keyphrases are noun phrases extracted from the document and ranked according to the sum of the significance of the words they contain. Although it improves the results, this scoring method has no proper justification and tends to assign high scores to long but non important phrases. For example, “nash equilibrium”, from the file *J-14.txt* of our evaluation dataset named SemEval, is a keyphrase composed of the two most significant words in the document, according to SingleRank. Therefore, SingleRank succeeds to extract it, but

candidates such as “unique nash equilibrium” or “exact nash equilibrium” which are longer, then have a better score, are extracted too. With TopicRank, we aim to circumvent this by ranking clusters of single and multi-word expressions instead of words.

Wan and Xiao (2008) use a small number of nearest neighbor documents to compute more accurate word co-occurrences and reinforce edge weights in the word graph. Borrowing co-occurrence information from multiple documents, their approach improves the word ranking performance. Instead of using words, Liang et al. (2009) use keyphrase candidates as vertices. Applied to Chinese, their method uses query log knowledge to determine phrase boundaries. Tsatsaronis et al. (2010) propose to connect vertices employing semantic relations computed using WordNet (Miller, 1995) or Wikipedia. They also experiment with different random walk algorithms, such as HITS (Kleinberg, 1999) or modified PageRank.

Liu et al. (2010) consider the topics of words using a Latent Dirichlet Allocation model (Blei et al., 2003, LDA). As done by Haveliwala (2003) for Information Retrieval, they propose to decompose PageRank into multiple PageRanks specific to various topics. A topic-biased PageRank is computed for each topic and corresponding word scores are combined. As this method uses a LDA model, it requires training data. With TopicRank, we also consider topics, but our aim is to use a single document, the document to be analyzed.

## 3 TopicRank

TopicRank is an unsupervised method that aims to extract keyphrases from the most important topics of a document. Topics are defined as clusters of similar keyphrase candidates. Extracting keyphrases from a document consists in the following steps, illustrated in Figure 1. First, the document is preprocessed (sentence segmentation, word tokenization and Part-of-Speech tagging) and keyphrase candidates are clustered into topics. Then, topics are ranked according to their importance in the document and keyphrases are extracted by selecting one keyphrase candidate for each of the most important topics.

Section 3.1 first explains how the topics are identified within a document, section 3.2 presents the approach we use to rank them and section 3.3 describes the keyphrase selection.

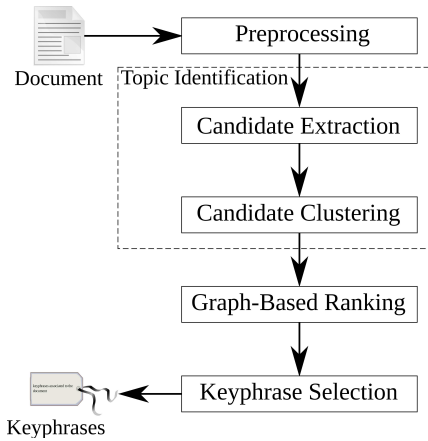


Figure 1: Processing steps of TopicRank.

### 3.1 Topic Identification

Keyphrases describe the most important topics of a document, thus the first step is to identify the keyphrase candidates that represent them. Hulth (2003) stated that most keyphrases assigned by human readers are noun phrases. Hence, the most important topics of a document can be found by extracting their most significant noun phrases. We follow Wan and Xiao (2008) and extract the longest sequences of nouns and adjectives from the document as keyphrase candidates. Other methods use syntactically filtered n-grams that are most likely to contain a larger number of candidates matching with reference keyphrases, but the n-gram restricted length is a problem. Indeed, n-grams do not always capture as much information as the longest noun phrases. Also, they are less likely to be grammatically correct.

In a document, a topic is usually conveyed by more than one noun phrase. Consequently, some keyphrase candidates are redundant in regard to the topic they represent. Existing graph-based methods (TextRank, SingleRank, etc.) do not take that fact into account. Keyphrase candidates are usually treated independently and the information about the topic they represent is scattered throughout the graph. Thus, we propose to group similar noun phrases as a single entity, a topic.

We consider that two keyphrase candidates are similar if they have at least 25% of overlapping words<sup>1</sup>. Keyphrase candidates are stemmed to reduce their inflected word forms into root forms<sup>2</sup>. To automatically group similar candidates into

<sup>1</sup>The value of 25% has been defined empirically.

<sup>2</sup>We chose to use stems because of the availability of stemmers for various languages, but using lemmas is another possibility that could probably work better.

topics, we use a Hierarchical Agglomerative Clustering (HAC) algorithm. Among the commonly used linkage strategies, which are complete, average and single linkage, we use the average linkage, because it stands as a compromise between complete and single linkage. In fact, using a highly agglomerative strategy such as complete linkage is more likely to group topically unrelated keyphrase candidates, whereas a strategy such as single linkage is less likely to group topically related keyphrase candidates.

### 3.2 Graph-Based Ranking

TopicRank represents a document by a complete graph in which topics are vertices and edges are weighted according to the strength of the semantic relations between vertices. Then, TextRank’s graph-based ranking model is used to assign a significance score to each topic.

#### 3.2.1 Graph Construction

Formally, let  $G = (V, E)$  be a complete and undirected graph where  $V$  is a set of vertices and the edges  $E$  a subset<sup>3</sup> of  $V \times V$ . Vertices are topics and the edge between two topics  $t_i$  and  $t_j$  is weighted according to the strength of their semantic relation.  $t_i$  and  $t_j$  have a strong semantic relation if their keyphrase candidates often appear close to each other in the document. Therefore, the weight  $w_{i,j}$  of their edge is defined as follows:

$$w_{i,j} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} \text{dist}(c_i, c_j) \quad (1)$$

$$\text{dist}(c_i, c_j) = \sum_{p_i \in \text{pos}(c_i)} \sum_{p_j \in \text{pos}(c_j)} \frac{1}{|p_i - p_j|} \quad (2)$$

where  $\text{dist}(c_i, c_j)$  refers to the reciprocal distances between the offset positions of the candidate keyphrases  $c_i$  and  $c_j$  in the document and where  $\text{pos}(c_i)$  represents all the offset positions of the candidate keyphrase  $c_i$ .

Our approach to construct the graph differs from TextRank.  $G$  is a complete graph and topics are therefore interconnected. The completeness of the graph has the benefit of providing a more exhaustive view of the relations between topics. Also, computing weights based on the distances between offset positions bypasses the need for a manually defined parameter, such as the window of words used by state-of-the-art methods (TextRank, SingleRank, etc).

<sup>3</sup> $E = \{(v_1, v_2) \mid \forall v_1, v_2 \in V, v_1 \neq v_2\}$



### Inverse problems for a mathematical model of ion exchange in a compressible ion exchanger

A mathematical model of ion exchange is considered, allowing for ion exchanger compression in the process of ion exchange. Two inverse problems are investigated for this model, unique solvability is proved, and numerical solution methods are proposed. The efficiency of the proposed methods is demonstrated by a numerical experiment.

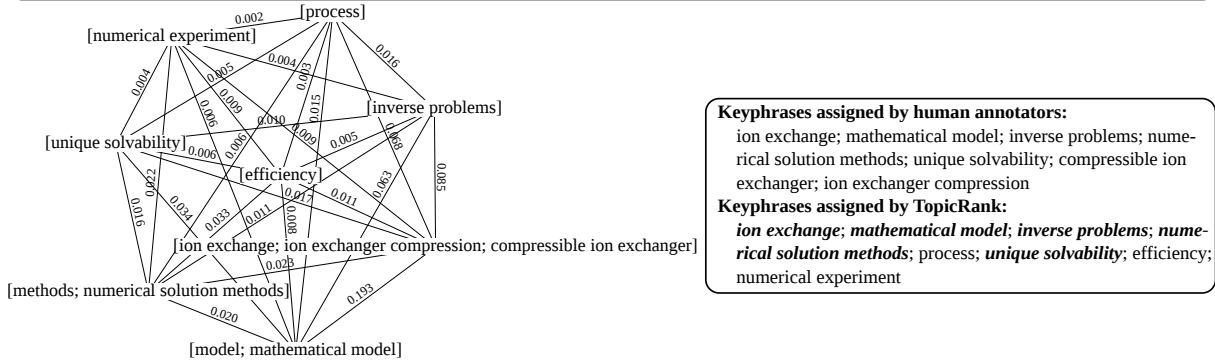


Figure 2: Sample graph build by TopicRank from Inspec, file 2040.abstr.

Figure 2 shows a sample graph built for an abstract from one of our evaluation datasets (Inspec). Vertices are topics, represented as clusters of lexically similar keyphrase candidates, and connected with all the others. In the example, we see the naivety of our clustering approach. Indeed, the clustering succeeds to group “ion exchanger”, “ion exchanger compression” and “compressible ion exchanger”, but the clustering of “methods” with “numerical solution methods” and “model” with “mathematical model” may be ambiguous as “methods” and “model” can be used to refer to other methods or models.

### 3.2.2 Subject Ranking

Once the graph is created, the graph-based ranking model TextRank, proposed by Mihalcea and Tarau (2004), is used to rank the topics. This model assigns a significance score to topics based on the concept of “voting”: high-scoring topics contribute more to the score of their connected topic  $t_i$ :

$$S(t_i) = (1 - \lambda) + \lambda \times \sum_{t_j \in V_i} \frac{w_{j,i} \times S(t_j)}{\sum_{t_k \in V_j} w_{j,k}} \quad (3)$$

where  $V_i$  are the topics voting for  $t_i$  and  $\lambda$  is a damping factor generally defined to 0.85 (Brin and Page, 1998).

### 3.3 Keyphrase Selection

Keyphrase selection is the last step of TopicRank. For each topic, only the most representative keyphrase candidate is selected. This selection avoids redundancy and leads to a good cover-

age of the document topics, because extracting  $k$  keyphrases precisely covers  $k$  topics.

To find the candidate that best represents a topic, we propose three strategies. Assuming that a topic is first introduced by its generic form, the first strategy is to select the keyphrase candidate that appears first in the document. The second strategy assumes that the generic form of a topic is the one that is most frequently used and the third strategy selects the centroid of the cluster. The centroid is the candidate that is the most similar to the other candidates of the cluster<sup>4</sup>.

## 4 Experimental Settings

### 4.1 Datasets

To compare the keyphrases extracted by TopicRank against existing methods, we employ four standard evaluation dataset of different languages, document sizes and domains.

The first dataset, formerly used by Hulth (2003), contains 2000 English abstracts of journal papers from the Inspec database. The 2000 abstracts are divided into three sets: a training set, which contains 1000 abstracts, a validation set containing 500 abstracts and a test set containing the 500 remaining abstracts. In our experiments we use the 500 abstracts from the test set. Several reference keyphrase sets are available with this dataset. Just as Hulth (2003), we use the uncontrolled reference, created by professional indexers.

The second dataset was built by Kim et al. (2010) for the keyphrase extraction task of the SemEval 2010 evaluation campaign. This dataset is

<sup>4</sup>The similarity between two candidates is computed with the stem overlap measure used by the clustering algorithm.

Corpus	Documents				Keyphrases		
	Type	Language	Number	Tokens average	Total	Average	Missing
Inspec	Abstracts	English	500	136.3	4913	9.8	21.8%
SemEval	Papers	English	100	5179.6	1466	14.7	19.3%
WikiNews	News	French	100	309.6	964	9.6	4.4%
DEFT	Papers	French	93	6844.0	485	5.2	18.2%

Table 1: Dataset statistics (missing keyphrases are counted based on their stemmed form).

composed of 284 scientific articles (in English) from the ACM Digital Libraries (conference and workshop papers). The 284 documents are divided into three sets: a trial set containing 40 documents, a training set, which contains 144 documents and a test set containing 100 documents. In our experiments we use the 100 documents of the test set. As for the reference keyphrases, we use the combination of author and reader assigned keyphrases provided by Kim et al. (2010).

The third dataset is a French corpus that we created from the French version of WikiNews<sup>5</sup>. It contains 100 news articles published between May 2012 and December 2012. Each document has been annotated by at least three students. We combined the annotations of each document and removed the lexical redundancies. All of the 100 documents are used in our experiments.

The fourth dataset is a French corpus made for the keyphrase extraction task of the DEFT 2012 evaluation campaign (Paroubek et al., 2012). It contains 468 scientific articles extracted from *Érudit*. These documents are used for two tasks of DEFT and are, therefore, divided in two datasets of 244 documents each. In our experiments we use the test set of the second task dataset. It contains 93 documents provided with author keyphrases.

Table 1 gives statistics about the datasets. They are different in terms of document sizes and number of assigned keyphrases. The Inspec and WikiNews datasets have shorter documents (abstract and news articles) compared to SemEval and DEFT that both contain full-text scientific articles. Also, the keyphrases provided with the datasets are not always present in the documents (less than 5% of missing keyphrases for Wikinews and about 20% of missing keyphrases for the other datasets). This induces a bias in the re-

sults. As explained by Hasan and Ng (2010), some researchers avoid this problem by removing missing keyphrases from the references. In our experiments, missing keyphrases have not been removed. However, we evaluate with stemmed forms of candidates and reference keyphrases to reduce mismatches.

## 4.2 Preprocessing

For each dataset, we apply the following preprocessing steps: sentence segmentation, word tokenization and Part-of-Speech tagging. For word tokenization, we use the TreebankWordTokenizer provided by the python Natural Language Toolkit (Bird et al., 2009) for English and the Bonsai word tokenizer<sup>6</sup> for French. For Part-of-Speech tagging, we use the Stanford POS-tagger (Toutanova et al., 2003) for English and MElt (Denis and Sagot, 2009) for French.

## 4.3 Baselines

For comparison purpose, we use three baselines. The first baseline is TF-IDF (Spärck Jones, 1972), commonly used because of the difficulty to achieve competitive results against it (Hasan and Ng, 2010). This method relies on a collection of documents and assumes that the  $k$  keyphrase candidates containing words with the highest TF-IDF weights are the keyphrases of the document. As TopicRank aims to be an improvement of the state-of-the-art graph-based methods for keyphrase extraction, the last two baselines are TextRank (Mihalcea and Tarau, 2004) and SingleRank (Wan and Xiao, 2008). In these methods, the graph is undirected, vertices are syntactically filtered words (only nouns and adjectives) and the edges are created based on the co-occurrences of words within a window of 2 for

<sup>5</sup>The WikiNews dataset is available for free at the given url: <https://github.com/adrien-bougouin/WikinewsKeyphraseCorpus>.

<sup>6</sup>The Bonsai word tokenizer is a tool provided with the Bonsai PCFG-LA parser: [http://alpage.inria.fr/statgram/frdep/fr\\_stat\\_dep\\_parsing.html](http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html).



Methods	Inspec			SemEval			WikiNews			DEFT		
	P	R	F	P	R	F	P	R	F	P	R	F
TF-IDF	32.7	38.6	33.4	13.2	8.9	10.5	33.9	35.9	34.3	10.3	19.1	13.2
TextRank	14.2	12.5	12.7	7.9	4.5	5.6	9.3	8.3	8.6	4.9	7.1	5.7
SingleRank	34.8	40.4	<b>35.2</b>	4.6	3.2	3.7	19.4	20.7	19.7	4.5	9.0	5.9
TopicRank	27.6	31.5	27.9	14.9	10.3	<b>12.1</b> <sup>†</sup>	35.0	37.5	<b>35.6</b> <sup>†</sup>	11.7	21.7	<b>15.1</b> <sup>†</sup>

Table 2: Comparison of TF-IDF, TextRank, SingleRank and TopicRank methods, when extracting a maximum of 10 keyphrases. Results are expressed as a percentage of precision (P), recall (R) and f-score (F). † indicates TopicRank’s significant improvement over TextRank and SingleRank at 0.001 level using Student’s t-test.

TextRank and 10 for SingleRank. As well as their window size, they differ in the weighting of the graph: TextRank has an unweighted graph and SingleRank has a graph weighted with the number of co-occurrences between the words. A graph-based ranking model derived from PageRank (Brin and Page, 1998) ranks each vertex and extracts multi-word keyphrases according to the ranked words. In TextRank, the  $k$ -best words are used as keyphrases and the adjacent sequences in the document are collapsed into multi-word keyphrases. Although  $k$  is normally proportional to the number of vertices in the graph, we set it to a constant number, because experiments conducted by Hasan and Ng (2010) show that the optimal value of the ratio depends on the size of the document. In SingleRank, noun phrases extracted with the same method as TopicRank are ranked by a score equal to the sum of their words scores. Then, the  $k$ -best noun phrases are selected as keyphrases.

For all the baselines, we consider keyphrase candidates which have the same stemmed form as redundant. Once they are ranked we keep the best candidate and remove the others. This can only affect the results in a positive way, because the evaluation is performed with stemmed forms, which means that removed candidates are considered equal to the retained candidate.

#### 4.4 Evaluation Measures

The performances of TopicRank and the baselines are evaluated in terms of precision, recall and f-score (f1-measure) when a maximum of 10 keyphrases are extracted ( $k = 10$ ). As said before, the candidate and reference keyphrases are stemmed to reduce the number of mismatches.

## 5 Results

To validate our approach, we designed three experiments. The first experiment compares TopicRank<sup>7</sup> to the baselines<sup>8</sup>, the second experiment individually evaluates the modifications of TopicRank compared to SingleRank<sup>9</sup> and the last experiment compares the keyphrase selection strategies. To show that the clusters are well ranked, we also present the results that could be achieved with a “perfect” keyphrase selection strategy.

Table 2 shows the results of TopicRank and the three baselines. Overall, our method outperforms TextRank, SingleRank and TF-IDF. The results of TopicRank and the baselines are lower on SemEval and DEFT (less than 16% of f-score), so we deduce that it is more difficult to treat long documents than short ones. On Inspec, TopicRank fails to do better than all the baselines, but on SemEval, WikiNews and DEFT, it performs better than TF-IDF and significantly outperforms TextRank and SingleRank. Also, we observe a gap between TF-IDF’s and the two graph-based baselines results. Although TopicRank is a graph-based method, it overcomes this gap by almost tripling the f-score of both TextRank and SingleRank.

Table 3 shows the individual modifications of TopicRank compared to SingleRank. We evaluate SingleRank when vertices are keyphrase candidates (+phrases), vertices are topics (+topics) and when TopicRank’s graph construction is used

<sup>7</sup>Results reported for TopicRank are obtained with the first position selection strategy.

<sup>8</sup>TopicRank and the baselines implementations can be found at the given url: [https://github.com/adrien-bougouin/KeyBench/tree/ijcnlp\\_2013](https://github.com/adrien-bougouin/KeyBench/tree/ijcnlp_2013).

<sup>9</sup>The second experiment is performed with SingleRank instead of TextRank, because SingleRank also uses a graph with weighted edges and is, therefore, closer to TopicRank.

Methods	Inspec			SemEval			WikiNews			DEFT		
	P	R	F	P	R	F	P	R	F	P	R	F
SingleRank	34.8	40.4	35.2	4.6	3.2	3.7	19.4	20.7	19.7	4.5	9.0	5.9
+phrases	21.5	25.9	22.1	9.6	7.0	8.0 <sup>†</sup>	28.6	30.1	28.9 <sup>†</sup>	10.5	19.7	13.5 <sup>†</sup>
+topics	26.6	30.2	26.8	14.7	10.2	11.9 <sup>†</sup>	31.0	32.8	31.4 <sup>†</sup>	11.5	21.4	14.8 <sup>†</sup>
+complete	34.9	41.0	<b>35.5</b>	5.5	3.8	4.4	20.0	21.4	20.3	4.4	9.0	5.8
TopicRank	27.6	31.5	27.9	14.9	10.3	<b>12.1<sup>†</sup></b>	35.0	37.5	<b>35.6<sup>†</sup></b>	11.7	21.7	<b>15.1<sup>†</sup></b>

Table 3: Comparison of the individual modifications from SingleRank to TopicRank, when extracting a maximum of 10 keyphrases. Results are expressed as a percentage of precision (P), recall (R) and f-score (F). <sup>†</sup> indicates a significant improvement over SingleRank at 0.001 level using Student’s t-test.

with word vertices (+complete). Using keyphrase candidates as vertices significantly improves SingleRank on SemEval, WikiNews and DEFT. On Inspec, it induces a considerable loss of performance caused by an important deficit of connections that leads to connected components, as shown in Figure 3. When we look at the distribution of “fuzzy” into the graph, we can see that it is scattered among the connected components and, therefore, increases the difficulty to select “fuzzy Bayesian inference techniques” as a keyphrase (according to the reference). The other datasets contain longer documents, which may dampen this problem. Overall, using topics as vertices performs better than using keyphrase candidates. Using topics significantly outperforms SingleRank on SemEval, WikiNews and DEFT. As for the new graph construction, SingleRank is improved on Inspec, SemEval and WikiNews. Results on DEFT are lower than SingleRank, but still competitive. Although the improvements are not significant, the competitive results point out that the new graph construction can be used instead of the former method, which requires to manually define a window of words. Experiments show that the three contributions are improvements and TopicRank benefits from each of them.

Table 4 shows the results of TopicRank when selecting either the first appearing candidate, the most frequent one or the centroid of each cluster. Selecting the first appearing keyphrase candidate is the best strategy of the three. It significantly outperforms the frequency and the centroid strategies on SemEval, WikiNews and DEFT. On SemEval and DEFT, we observe a huge gap between the results of the first position strategy and the others. The two datasets are composed of scientific articles where the full form of the main topics are

often introduced at the beginning and then, conveyed by abbreviations or inherent concepts (e.g. the file *C-17.txt* from SemEval contains *packet-switched network* as a keyphrase where *packet* is more utilized in the content). These are usually more similar to the generic form and/or more frequent, which explains the observed gap.

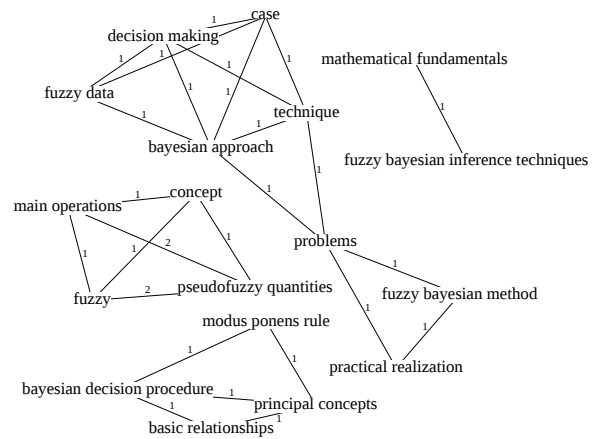


Figure 3: Connected component problem with the method SingleRank+phrases. Example taken from Inspec, file *1931.abstr*.

To observe the ranking efficiency of TopicRank, we also evaluate it without taking the keyphrase selection strategy into account. To do so, we extract the top-ranked clusters and mark the reference keyphrases into them. We deduce the upper bound results of our method by computing the precision, recall and f-score where the number of correct matches is equal to the number of clusters containing at least one reference keyphrase. The upper bound results show that our method could possibly perform better than all the baselines for the four datasets. Even on Inspec, the loss of performance can be bypassed by a more efficient keyphrase selection strategy.

Methods	Inspec			SemEval			WikiNews			DEFT		
	P	R	F	P	R	F	P	R	F	P	R	F
First position	27.6	31.5	27.9	14.9	10.3	12.1 <sup>†</sup>	35.0	37.5	35.6 <sup>†</sup>	11.7	21.7	15.1 <sup>†</sup>
Frequency	26.7	30.2	26.8	1.7	1.2	1.4	25.7	27.6	26.2	1.9	3.8	2.5
Centroid	24.5	28.0	24.7	1.9	1.2	1.5	28.1	29.9	28.5	2.6	5.0	3.4
Upper bound	36.4	39.0	<b>35.6</b>	37.6	25.8	<b>30.3</b>	42.5	44.8	<b>42.9</b>	14.9	28.0	<b>19.3</b>

Table 4: Comparison of the keyphrase candidate selection strategies against the best possible strategy (upper bound), when extracting a maximum of 10 keyphrases. Results are expressed as a percentage of precision (P), recall (R) and f-score (F). † indicates the first position strategy’s significant improvement over the frequency and the centroid strategies at 0.001 level using Student’s t-test.

## 6 Conclusion and Future Work

In this paper we presented TopicRank, an unsupervised method for keyphrase extraction. TopicRank extracts the noun phrases that represent the main topics of a document. The noun phrases are clustered into topics and used as vertices in a complete graph. The resulting graph stands as a topical representation of the document. Topics are scored using the TextRank ranking model and keyphrases are then extracted by selecting the most representative candidate from each of the top-ranked topics. Our approach offers several advantages over existing graph-based keyphrase extraction methods. First, as redundant keyphrase candidates are clustered, extracted keyphrases cover the main topics of the document better. The use of a complete graph also captures the relations between topics without any manually defined parameters and induces better or similar performances than the state-of-the-art connection method that uses a co-occurrence window. We conducted experiments on four standard evaluation datasets of different languages, document sizes and domains. Results show that TopicRank outperforms TF-IDF and significantly improves the state-of-the-art graph-based methods on three of them.

In future work, we will further improve the topic identification and the keyphrase selection. More precisely, we will develop an evaluation process to determine cluster quality and then focus on experimenting with other clustering algorithms and investigate the use of linguistic knowledge for similarity measures. As for the keyphrase selection, our experiments show that the current method does not provide the best solution that could be achieved with the ranked clusters. We plan to improve it using machine learning methods.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their useful advice and comments. This work was supported by the French National Research Agency (TermITH project – ANR-12-CORD-0029).

## References

- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O’Reilly Media.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1):107–117.
- Pascal Denis and Benoît Sagot. 2009. Coupling an Annotated Corpus and a Morphosyntactic Lexicon for State-of-the-Art POS Tagging with Less Human Effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pages 110–119, Hong Kong, December. City University of Hong Kong.
- Juhyun Han, Taehwan Kim, and Joongmin Choi. 2007. Web Document Clustering by Using Automatic Keyphrase Extraction. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 56–59, Washington, DC, USA. IEEE Computer Society.
- Kazi Saidul Hasan and Vincent Ng. 2010. Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 365–373, Stroudsburg, PA, USA. Association for Computational Linguistics.

- Taher H. Haveliwala. 2003. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796.
- Anette Hulth. 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM)*, 46(5):604–632, sep.
- Weiming Liang, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2009. Extracting Keyphrases from Chinese News Articles Using TextRank and Query Log Knowledge. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pages 733–740, Hong Kong, December. City University of Hong Kong.
- Marina Litvak and Mark Last. 2008. Graph-Based Keyword Extraction for Single-Document Summarization. In *Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization*, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to Find Exemplar Terms for Keyphrase Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, pages 257–266, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic Keyphrase Extraction Via Topic Decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Olena Medelyan and Ian H. Witten. 2008. Domain-Independent Automatic Keyphrase Indexing with Small Training Sets. *Journal of the American Society for Information Science and Technology*, 59(7):1026–1040, may.
- Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order Into Texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.
- George A. Miller. 1995. WordNet: a Lexical Database for English. *Communications of the Association for Computational Linguistics*, 38(11):39–41.
- Patrick Paroubek, Pierre Zweigenbaum, Dominic Forest, and Cyril Grouin. 2012. Indexation libre et contrôlée d’articles scientifiques. Présentation et résultats du défi fouille de textes DEFT2012 (Controlled and Free Indexing of Scientific Papers. Presentation and Results of the DEFT2012 Text-Mining Challenge) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, Workshop DEFT 2012: DÉfi Fouille de Textes (DEFT 2012 Workshop: Text Mining Challenge)*, pages 1–13, Grenoble, France, June. ATALA/AFCP.
- Karen Spärck Jones. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21.
- Takashi Tomokiyo and Matthew Hurst. 2003. A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, pages 33–40, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- George Tsatsaronis, Iraklis Varlamis, and Kjetil Nørvåg. 2010. SemanticRank: Ranking Keywords and Sentences Using Semantic Graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1074–1082, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, pages 855–860. AAAI Press.
- Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries*, pages 254–255, New York, NY, USA. ACM.

# Understanding the Semantic Intent of Domain-Specific Natural Language Query

Juan Xu, Qi Zhang, Xuanjing Huang

Fudan University

School of Computer Science

{11210240066, qz, xjhuang}@fudan.edu.cn

## Abstract

Queries asked on search engines nowadays increasingly fall in full natural language, which refer to Natural Language queries (NL queries). Parsing that kind of queries for the purpose of understanding user's query intent is an essential factor to search engine. To this end, a hierarchical structure is introduced to represent the semantic intent of NL query and then we focus on the problem of mapping NL queries to the corresponding semantic intents. We propose a parsing method by conducting two steps as follows: (1) predicting semantic tags for a given input query; (2) building an intent representation for the query using the sequence of semantic tags based on a structured SVM classification model. Experimental results on a manually labeled corpus show that our method achieved a sufficiently high result in term of precision and F1.

## 1 Introduction

Nowadays, with the development of voice search, queries asked on search engines often fall in full natural language, which refer to NL queries. For example, for the purpose of looking for a restaurant, it is natural for us to ask "find the best Italian restaurant near seattle washington" rather than "Italian restaurant seattle wa". This means that voice search users are liable to express their intent in natural language which differs significantly from Web users. In addition to the developing of voice search, the increasing use of smartphones with voice assistant further boosts the number of such natural language queries.

This increasing amount of natural language queries brings a big challenge to search engines. Many search engines today, generally speaking,

are based on matching keywords against structured information from relational database. Consider the query "find the best Italian restaurant near seattle washington". Without understanding the semantic intent, it may retrieval some unsatisfying results that merely contain all these keywords, which are not really search terms (e.g. restaurant). But when the search engine understands the intent of this query is to "find restaurant", and also knows the meanings of individual constituents (i.e the "restaurant" is head search terms, "the best Italian" and "near seattle washington" are modifier), then it would be able to route the query to a specialized search module (in this case restaurant search) and return the most relevant and essential answers rather than results that merely contain all these keywords.

In no small part, the success of such approach relies on robust understanding of query intent. Most previous works in this area focus on query tagging problem, i.e., assigning semantic labels to query terms (Li et al., 2009; Manshadi and Li, 2009; Sarkas et al., 2010; Bendersky et al., 2011). Indeed, with the label information, a search engine is able to provide users with more relevant results. But previous works have not considered the issue for understanding the semantic intent of NL queries and their methods are not suitable for interpreting the semantic intent of this kind of complex queries. In this work, in order to enable search engines to understand natural language query, we focus on the problem of mapping NL queries from a particular search engine like Google maps, Bing maps etc, to their semantic intents representation. A key contribution of this work is that we formally define a hierarchical structure to represent the semantic intents. As an example, consider a query about finding a local business near some location such as:

**Example** : *find the best Italian restaurant near space needle seattle washington*

This query has four constituents: the Business that the user is looking for (restaurant), the Neighborhood (space needle seattle washington), the condition and cuisine type that the user specified (the best Italian) and the term that helps user to ask (find). To understand the semantic intent of the query, the model should not only be able to recognize the four constituents but also needs to understand the structure of each constituent. Therefore we are looking for a model that is able to generate the semantic intent representation for this query as shown in Figure 1.

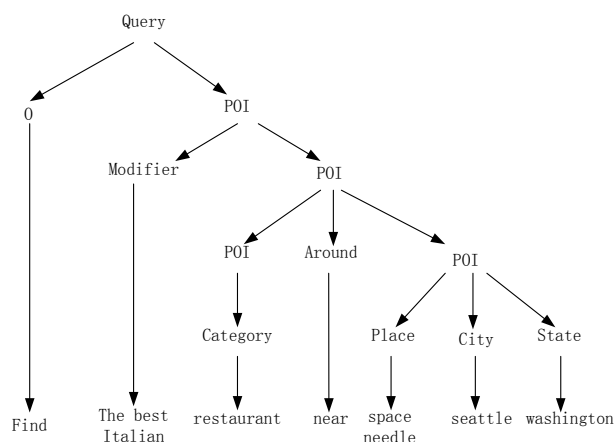


Figure 1: A simple grammar tree for local businesses search.

Generating the hierarchical structure of queries can be beneficial to information retrieval. Knowing the semantic role of each query constituent, we can reformulate the query into a structured form or reweight different query constituents for structured data retrieval (Kim et al., 2009; Paparizos et al., 2009; Gollapudi et al., 2011; Kim and Croft, 2012). Alternatively, the knowledge of the structure of the constituents helps route the query to a specialized search module.

The second contribution of this work is that we define a grammar which is isomorphic to a Context Free Grammar (CFG), and also present an approach which can automatically generate the semantic intent representation for NL queries by considering it as a structure classification problem.

The rest of this paper are organized as follows. Section 2 details the related work. In Section 3 we demonstrate the hierarchical structured representation of queries and introduce our grammar for it. Then, in Section 4, we propose our method for parsing the queries. Section 5 presents the experi-

mental results. We draw the conclusions from our work in Section 6.

## 2 Related Works

### 2.1 Query Intent Understanding

To capture the underlying information need encoded within diverse user queries, considerable works have been conducted from many aspects. Most previous works in this area focus on query intent classification as one aspect, i.e., automatically mapping queries into semantic classes or patterns (Li et al., 2008; Arguello et al., 2009; Duan et al., 2012). There is another aspect to study the problem, query tagging, i.e., assigning semantic labels to query terms (Li et al., 2009; Sarkas et al., 2010; Bendersky et al., 2011). For example, in “Restaurant Rochester Chinese MN”, the word “Restaurant” should be tagged as *Business*. In particular, Li et al. leverage clickthrough data and a database to automatically derive training data for learning a CRF-based tagger. Manshadi and Li develop a hybrid, generative grammar model for a similar task. Sarkas et al. consider an annotation as a mapping of a query to a table of structured data and attributes of this table, while Bendersky et al. mark up queries with annotations such as part-of-speech tags, capitalization, and segmentation.

There are relatively little published work on understanding the semantic intent of natural language query. Manshadi and Li (2009) and Li (2010) consider the semantic structure of queries. In particular, Li (2010) defines the semantic structure of noun-phrase queries as intent heads (attributes) coupled with some number of intent modifiers (attribute values), e.g., the query [alice in wonderland 2010 cast] is comprised of an intent head *cast* and two intent modifiers *alice in wonderland* and *2010*. Our approach differs from the earlier work in that we investigate the natural language query intent understanding problem, and build a hierarchical representation for it.

### 2.2 Semantic Parsing

For the purpose of enabling search engines to understand user’s query intent, we present an approach to parse NL queries to the corresponding semantic intents, which is similar to the task in semantic parsing (Kate et al., 2005). We parse the queries to a hierarchical structure consisting of all query terms. While the task of semantic parsing is mapping NL input to its interpretation ex-

pressed in well-defined formal *meaning representation*(MR) language. For example, “How many states does the Colorado river run through?”, the output of semantic parsing is “count( state( traverse( river( const(colorado))))”. Although it can express the meanings of NL inputs, it is not suitable for search engines to use, which, generally speaking, are based on matching keywords against documents.

### 3 Query Intent Representation and Grammar for NL Query

The task we defined is mapping the natural language query (NL query) to the corresponding intent representation(IR), which is specifically for a particular search scenario like Google maps and Bing maps etc, but can generalize well to other search scenarios by redefining the grammar.

#### 3.1 Query Intent Representation

In this work, we propose to use a tree structure to represent the semantic intent of a query. Consider the instance in section 1, the NL query “find the best Italian restaurant near space needle seattle washington” consists of 10 words. The IR is a hierarchical tree structure, as shown in Figure 1.

There are 9 different non-terminal symbols in the tree, of which Query is the start symbol. And it contains 11 rules to formulate the tree structure. As shown in the Figure 1, it clearly depicts that the query has 2 constituents and also depicts the structure of each constituent. After having that tree structure, search engines can easily understand the semantic intent of the query as we discussed in introduction.

#### 3.2 Grammar for NL Query

We have manually designed a grammar for the purpose of automatic generating the hierarchical structure of queries. As mentioned above, we focus on a particular search scenario (i.e map search domain). Based on analysis of NL queries from that domain, we observe that most queries carry an underlying structure. Therefore a set of CFG rules were written for the map search scenario. Below

are some sample rules from those CFG rules:

Query -- > O POI Around POI  
 POI -- > Place IN City State  
 POI -- > Num Road IN City State  
 POI -- > Road  
 POI -- > Category IN City  
 Query -- > Modifier Transition From POI TO POI

And we also define a set of semantic tags for that kind of queries which indicate the semantic role of each query constituent. More formally we define a Context-Free Grammar (CFG) for NL query as a 4-tuple  $G=(N, T, S, R)$  where

- $N$  is a set of non-terminals;
- $T$  is a set of terminals;
- $S \in N$  is a special non-terminal called start symbol,
- $R$  is a set of rules  $\{A \rightarrow \beta\}$  where  $A$  is non-terminal and  $\beta$  is a string of symbols from the infinite set of strings of  $(N \cup T)^*$ .

The sequence of  $\implies$  used to derive  $w$  from  $S$  is called a derivation of  $w$ . Here  $\implies^*$  is defined as the reflexive transitive closure of  $\implies$ . We can then formally define the language  $L(G)$  generated by the grammar  $G$  as the set of strings composed of terminal symbols that can be drawn from the designated start symbol  $S$ .

$$L = \{w | w \text{ is in } T^* \text{ and } S \implies^* w\}$$

Given the above definitions, parsing a string  $w$  means to find all (if any) the derivations of  $w$  from  $S$ .

Our grammar composes a constraint ordering on queries. And it is reasonable, because the query we investigated is full natural language query. While there are some NL queries whose words sequences are arbitrary, which is not in our consideration.

## 4 Methodology

To produce the semantic intent representation for an NL query from map search domain, we need to extract the basic semantic query constituents and build the semantic intent representation with them according to the query. Summarily, the input of our parsing task is a NL query and the output is the intent representation. The proposed method for this problem consists of two phases:

Semantic category	Meaning	Example
O	The useless information for search	where is
M	The modifier which indicates user's personal interests	the best and cheapest
F	The information of predicting the start address in direction search	from
Q	The information of predicting the end address in direction search	to
C	The city	New York
S	The state	North Carolina(Abbr. NC)
P	The place	White House
T	The town	Forbes
A	The predicted information of the area range of a point	nearby
L	The POI category	restaurant
I	The predicted information of a point within an area	in
D	The district	Brooklyn
N	The number of an address	Room 606
R	The road	Church St
G	The country	USA
W	The transition	via subway

Table 1: Illustration for each semantic category.

1. Predicting a sequence of semantic tags for a given input sentence.
2. Building an intent representation with the sequence of semantic tags

We describe a method using CRFs and structured support vector machine (SSVMs) in the following subsection for the first step and the second step, respectively.

#### 4.1 Semantic Tagging with CRFs

Let  $w_1w_2\dots w_N$  be a NL sentence in which  $N$  is the number of words and  $w_i$  is  $i^{th}$  word. Assuming that a chunk tag for a sequence of words  $w_i\dots w_j(1\leq i\leq j\leq N)$  is  $t_i$ , the lexical semantic prediction problem is to determine a lexical semantic tag  $s_i$  for a sequence of words  $w_i\dots w_j(1\leq i\leq j\leq N)$ . In the meantime, semantic tag  $s_i$  is structural and consists of two parts:

1) **Boundary Category:**  $BC = \{B, I\}$ . Here B/I means that current word is at the beginning/in the middle of a semantic chunk.

2) **Semantic Category:**  $SC = \{O, M, F, Q, C, S, P, T, A, L, I, D, N, R, G, W\}$ . This is used to denote the class of the semantic name. The meaning of each semantic class name represented is illustrated in Table 1.

The conditional random fields CRFs (Lafferty et al., 2001) have shown empirical successes in label sequence labelling problem. Therefore, we exploit

the use of CRFs to our semantic tagging problem in which a feature set for this task is presented in Table 2.

#### 4.2 Generating Intent Representation with Structural SVMs

This subsection first gives some background about the SSVMs for structured prediction, and then we focus on how to use SSVMs to our intent representation learning problem.

##### 4.2.1 Structural SVMs

Suppose a training set of input-output structure pairs  $S = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (X \times Y)^n$  is given. Structured classification is the problem of predicting  $y$  from  $x$  in the case where  $y$  has a meaningful internal structure. Elements  $y \in Y$  may be, for instance, sequences, strings, labeled trees, lattices, or graphs.

The approach we pursue is to learn a discriminant function  $F : X \times Y \rightarrow R$  over  $S$ . For a specific given input  $x$ , we can derive a prediction by maximizing  $F$  over the response variable. Hence, the general form of our hypotheses  $f$  is

$$f(x; w) = \operatorname{argmax}_{y \in Y} F(x; y; w) \quad (1)$$

where  $w$  denotes a parameter vector.

As the principle of the maximum-margin presented in (Vapnik, 1995), in the structured classification problem, Tsochantaridis (2004) proposed several maximum-margin optimization problems



Feature index	Definition
1	The current word and the preceding word
2	The current word and the following word
3	The current word
4	The knowledge of Pre-1 word in Geo-Knowledge Database
5	The knowledge of Pre-2 word in Geo-Knowledge Database
6	The knowledge of Post-1 word in Geo-Knowledge Database
7	The knowledge of Post-2 word in Geo-Knowledge Database
8	The knowledge of current word in Geo-Knowledge Database

Table 2: Features for CRFs Model.

$\delta\psi_i(y) \equiv \psi(x_i, y_i) - \psi(x_i, y)$ . The soft-margin criterion was proposed in order to allow errors in the training set, by introducing slack variables.

$$\text{SVM}_1 : \underbrace{\min_w}_{w} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \quad s.t. \forall i, \xi_i \geq 0 \quad (2)$$

$$\forall i, \forall y \in Y \setminus y_i : \langle w, \delta\psi_i(y) \rangle > 1 - \xi_i \quad (3)$$

Alternatively, using a quadratic term  $\frac{C}{2n} \sum_i \xi_i^2$  to penalize margin violations,  $\text{SVM}_2$  would be obtained. Here  $C > 0$  is a constant that control the tradeoff between training error minimization and margin maximization.

To deal with problems in which  $|Y|$  is very large, such as Natural Language parsing, Tsochantaridis (2004) proposed two approaches that generalize the formulation  $\text{SVM}_1$  and  $\text{SVM}_2$  to the cases of arbitrary loss function. We use  $\text{SVM}_1$  to introduce that two approaches and they are also work for  $\text{SVM}_2$ . The first approach is to rescale the slack variables according to the loss incurred in each of the linear constraints.

$$\text{SVM}_1^{\Delta_s} : \underbrace{\min_w}_{w} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \quad s.t. \forall i, \xi_i \geq 0 \quad (4)$$

$$\forall i, \forall y \in Y \setminus y_i : \langle w, \delta\psi_i(y) \rangle \geq \frac{1 - \xi_i}{\Delta(y_i, y)} \quad (5)$$

The second approach to include loss function is to rescale the margin as a special case of the Hamming loss. The margin constraints in this setting take the following form:

$$\forall i, \forall y \in Y \setminus y_i : \langle w, \delta\psi_i(y) \rangle \geq \Delta(y_i, y) - \xi_i \quad (6)$$

This set of constraints yields an optimization problem, namely  $\text{SVM}_1^{\Delta^m}$ .

The algorithm to solve the maximum-margin problem in structured learning problem is presented in detail in (Tsochantaridis et al., 2004). And

it can be applied to all SVM formulations mentioned above. The only difference between them is the cost function.

Following the successes of the Structural SVMs algorithm to structured prediction, we exploit the use of SSVM to our parsing task. As discussed in (Tsochantaridis et al., 2004), the major problem to apply the Structural SVMs is to implement the feature mapping  $\psi(x, y)$ , the loss function  $\Delta(y_i, y)$ , as well as the maximization algorithm. In the following section, we apply a Structural SVMs to the problem of semantic intent learning in which the mapping function, the maximization algorithm, and the loss function are introduced.

#### 4.2.2 Feature mapping

For our task, we can choose a mapping function to get a model that is isomorphic to a probabilistic grammar in which each rule within the grammar is defined by our own based on the application area. Each node in a parse tree  $y$  for an NL query  $x$  corresponds to a grammar rule  $g_j$ , which in turn has a score  $w_j$ .

All valid parse trees  $y$  for an NL query  $x$  are scored by the sum of the  $w_j$  of their nodes, and the feature mapping  $\psi(x, y)$  is a histogram vector counting how often each grammar rule  $g_j$  occurs in the tree  $y$ . The example shown in Figure 2 clearly depicts the way features are mapped from a tree structure intent representation of an NL query.

#### 4.2.3 Loss function

Typically, the correctness of a predicted parse tree is measured by its F1 score (see e.g. (Johnson, 1998)), the harmonic mean of precision of recall as calculated based on the overlap of nodes between the trees. In this work, we follow this loss function and introduce the standard zero-one classification loss as a baseline measure method.

Let  $z$  and  $z_i$  be two parse tree outputs and  $|z|$  and  $|z_i|$  be the number of brackets in  $z$  and  $z_i$ , re-

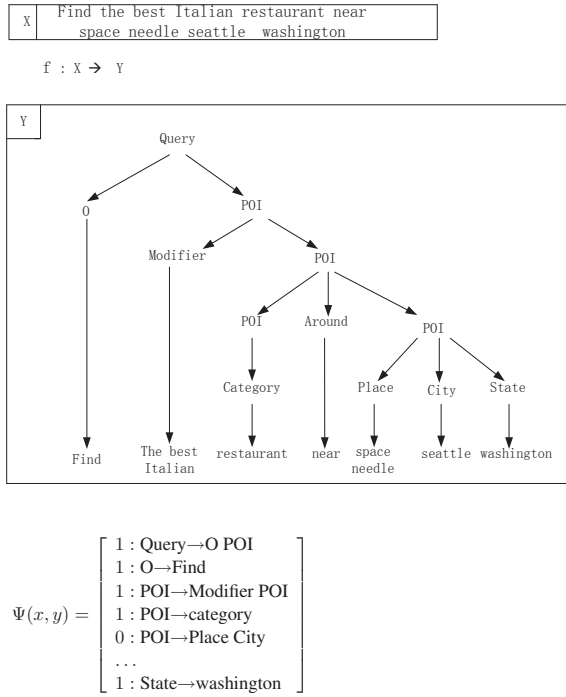


Figure 2: Example of feature mapping using tree representation.

spectively. Let  $n$  be the number of common brackets in the two trees. The loss function between  $z_i$  and  $z$  is computed as bellow.

$$F - \text{loss}(z_i, z) = 1 - \frac{2 \times n}{|z| + |z_i|} \quad (7)$$

$$\text{zero} - \text{one}(z_i, z) = \begin{cases} 1 & \text{if } z_i \neq z \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

#### 4.2.4 Maximization algorithm

Note that the learning function can be efficiently computed by finding the structure  $y \in Y$  that maximizes  $F(x; y; w) = \langle w, \delta\psi_i(y) \rangle$  via a maximization algorithm. To this end, we use a modified version of the CKY parser of Mark Johnson<sup>1</sup> and incorporated it into our algorithm.

#### 4.2.5 SSVM learning algorithm

Algorithm 1 shows our generation of SSVM learning for the semantic intent representation learning problem. The algorithm can apply to all SVM formulations discussed in section 4.2.1. The only difference is in the way the cost function gets set up in step 3 and the other three optional cost function are:

$$\text{SVM}_2^{\Delta^s} : H(y) \equiv (1 - \langle \delta\psi_i(y), w \rangle) \sqrt{\Delta(y_i, y)}$$

<sup>1</sup><http://www.cog.brown.edu/~mj/Software.htm>.

### Algorithm 1 Algorithm of SSVM learning for query parsing

**Input:**  $I = (x_i; y_i), i = 1, 2, \dots, l$  in which  $x_i$  is the NL query's semantic tags sequence and  $y_i$  is the corresponding tree structure.

**Output:** SSVM model

- 1: **repeat**
- 2:   **for**  $i = 1$  **to**  $l$  **do**
- 3:     set up cost function based on the corresponding optimization problem;  
 $\text{SVM}_1^{\Delta^s} : H(y) \equiv (1 - \langle \delta\psi_i(y), w \rangle) \Delta(y_i, y)$
- 4:     compute  $\hat{y} = \text{argmax}_{y \in Y} H(y)$
- 5:     compute  $\xi_i = \max\{0, \max_{y \in S_i} H(y)\}$
- 6:     **if**  $H(\hat{y}) > \xi_i + \epsilon$  **then**
- 7:        $S_i \leftarrow S_i \cup \{\hat{y}\}$
- 8:     solving optimization with SVM;
- 9:     **end if**
- 10:  **end for**
- 11: **until** no  $S_i$  has changed during iteration

$$\text{SVM}_1^{\Delta^m} : H(y) \equiv (\Delta(y_i, y) - \langle \delta\psi_i(y), w \rangle)$$

$$\text{SVM}_2^{\Delta^m} : H(y) \equiv (\sqrt{\Delta(y_i, y)} - \langle \delta\psi_i(y), w \rangle)$$

The feature mapping  $\psi(x, y)$ , the loss function  $\Delta(y_i, y)$ , as well as the maximization in step 6 were implemented as mentioned in above. A working set  $S_i$  is maintained for each training example  $(x_i, y_i)$  to keep track of the selected constraints which define the current relaxation. The algorithm stops, if no constraint is violated by more than  $\epsilon$  and then we get a SSVM model. Note that the SVM optimization problems from iteration to iteration differ only by a single constraint. We therefore restart the SVM optimizer from the current solution, which really reduces the runtime.

## 5 Experiments

### 5.1 Corpus

To facilitate the study we need benchmark corpus with ground-truth semantic intent representations in map search area. Since there is no such kind of corpus publicly available, we constructed a corpus MSIntent from answers.yahoo.com, which have a large number of queries submitted by users. The MSIntent corpus contains 1200 NL queries. Since those queries were crawled from an open question domain which contains many noises, 670 NL queries were finally chosen and manually labeled.

Two annotators labeled the corpus independently. The annotators first tagged each query with a set of semantic tags, and then build it's corre-

sponding semantic intent representation tree based on a given grammar, which consists of 17 non-terminal and 269 productions, defined specifically for our task. In order to keep the reliability of annotations, another annotator was asked to check the corpus and determine the conflicts. Finally we got a corpus includes 670 NL queries and their corresponding semantic intent representation. Table 3 shows the statistic on our corpus MSIntent.

Statistic	Numbers
No.of. Examples	670
Avg. NL sentence length	10.11
No. of non-terminals	17
No. of productions	227

Table 3: Statistics on MSIntent corpus. The average length of an NL query in the corpus is 10.11 words. This indicates that MSIntent is the hard corpus.

## 5.2 Experiments Configurations

We use the standard 10-fold cross validation test for evaluating the method. NL test queries were first tagged with a sequence of semantic tags, and those sequences were used to build trees, which has been detailed in section 3, via a structured support vector machine.<sup>2</sup> We evaluate the accuracy of tagging an NL query to a sequence of semantic tags by computing the total number of correct semantic tags in comparison with the gold-standard. For this purpose, CRF model<sup>3</sup> obtains a high result, with 91.73% accuracy.

Our main focus is to evaluate the proposed method in parsing NL queries to IR, we measure the number of test queries that produced complete IR, and the number of those IR that were correct. For our task, a IR is correct if it exactly matches the correct representation, we use the evaluation method for semantic parsing problem presented in (Kate et al., 2005) as the formula be:

$$\text{Precision} = \frac{\#correct\ IR}{\#completed\ IR}$$

$$\text{Recall} = \frac{\#correct\ IR}{\#queries}$$

$$\text{F1} = \frac{2Precision \cdot Recall}{Precision + Recall}$$

<sup>2</sup><http://svmlight.joachims.org/>

<sup>3</sup>We use CRF++ toolkit, <http://crfpp.sourceforge.net/>

## 5.3 Results

In this section we show that with high tagging accuracy as mentioned above, our proposed method for parsing NL queries to IR is effective. To this end, we conducted two sets of experiment. The first experiment was to show the performance of our proposed method in terms of precision, recall and F-score measurements. The second was to investigate the effect of other kernel functions in our learning algorithm.

**1. Performance of our method.** The results are given in Table 4, which shows recall, precision and F1 for test set. The first line shows the performance for PCFG model as trained on our MSIntent corpus by Johnson’s implementation. The following two lines show the  $SVM_2^{\Delta^m}$  and  $SVM_2^{\Delta^s}$  with zero-one loss, while the rest lines give the results for the F1-loss. All results are for  $C = 39$  and  $\epsilon = 0.1$ . All values of  $C$  between 0.1 to 100 gave comparable results. Table 4 indicates that our method achieved high accuracy results.  $SVM_2^{\Delta^m}$  using the F-loss gives better F1, outperforming the PCFG substantially. We conjecture that we can achieve further gains by incorporating more complex features into the grammar, which would be impossible or at least awkward to use in a PCFG model.

**2. The effect of kernel functions in our learning algorithm.** As seen in Table 5, it shows the training and testing results for various kernel functions including linear kernel, polynomial kernel, and RBF kernel. The regularization parameter  $C$  and the criterion parameter  $\epsilon$  are set to the same values as that in the first experiment. From the results,  $SVM_2^{\Delta^s}$  with polynomial kernel obtains 83.86% recall, 89% precision and 86.43% F1, which is the best result. But we observe that our proposed method can perform well with different kernel functions without significantly difference.

In addition, when performing SSVM on the test set, we might obtain some ‘NULL’ outputs since the grammar generated by SSVM could not derive this sentence, but generally we obtained high recall. Summarily, Table 5 depicts that the proposed method using different parameters achieved high performance in term of precision.

## 6 Conclusions

This paper presented a new facet to investigate the semantic intent of NL queries, which maps N-L queries to the corresponding semantic intents.

Parameter	Test Recall	Test Precision	Test F1
<b>PCFG</b>	79.10	89.83	84.12
<b>0/1-loss(SVM<math>\Delta_2^m</math>)</b>	83.43	88.05	85.78
0/1-loss(SVM $\Delta_2^s$ )	83.26	88.57	85.47
<b>F-loss(SVM<math>\Delta_2^m</math>)</b>	<b>83.42</b>	<b>88.72</b>	<b>85.97</b>
F-loss(SVM $\Delta_2^s$ )	83.01	88.39	85.60

Table 4: Results for parsing NL queries to IR on MSIntent corpus using cross-validation test.

Parameter	Training Accuracy	Test Recall	Test Precision	Test F1
linear+F-loss( $\Delta_m$ )	92.37	83.42	88.72	85.97
polynomial(d=2)+ F-loss( $\Delta_m$ )	91.66	82.98	88.23	85.13
<b>polynomial(d=2)+ F-loss(<math>\Delta_s</math>)</b>	<b>91.98</b>	<b>83.86</b>	<b>89.00</b>	<b>86.43</b>
RBF+F-loss( $\Delta_m$ )	92.00	83.11	88.22	85.17
RBF+F-loss( $\Delta_s$ )	92.34	82.67	87.96	84.92

Table 5: Experiment results for parsing NL queries to IR on MSIntent corpus using SSVMs with various kernel functions.

We also proposed a method of using a hierarchical structure to represent the semantic intent of N-L query, and then presented an automatic method to learn the semantic intent representation with the corpus of NL queries and their semantic intent representation in tree structure.

Experimental results with a manually labeled corpus have demonstrated that our method achieves a very good performance in term of precision and F1-scores. We thus can confidently conclude that the structured support vector models are suitable to the problem of our semantic intent learning problem. We also provide a semantic tagging tool with a very high accuracy by using a CRF model that can be beneficially used as pre-processing for the semantic intent learning problem.

The main drawback with our approach is that we strict the ordering of NL queries. Note that although strict ordering constraints such as those imposed by CFG is appropriate for modeling query structure in our task, it might be helpful to somehow ignore the ordering. We leave this for future work. Another interesting and practically useful problem that we have left for future work is

to extend our method to a version of SVM semi-supervised learning. Having such a capability, we are able to automatically learn the semantic intent of NL queries by processing labeled and unlabeled data.

## 7 Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (61003092, 61073069), National Major Science and Technology Special Project of China (2014ZX03006005), Shanghai Municipal Science and Technology Commission (No.12511504500) and ‘‘Chen Guang’’ project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation(11CG05).

## References

Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. 2009. Sources of evidence for vertical selection. In *Proceedings of the 32nd international ACM SIGIR conference on Research and*

- development in information retrieval*, pages 315–322. ACM.
- Michael Bendersky, W Bruce Croft, and David A Smith. 2011. Joint annotation of search queries. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 102–111.
- Huizhong Duan, Emre Kiciman, and ChengXiang Zhai. 2012. Click patterns: an empirical representation of complex query intents. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1035–1044. ACM.
- Sreenivas Gollapudi, Samuel Jeong, Alexandros Ntoulas, and Stelios Paparizos. 2011. Efficient query rewrite for structured web queries. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2417–2420. ACM.
- Mark Johnson. 1998. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.
- Rohit J Kate, Yuk Wah Wong, and Raymond J Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1062. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Jin Young Kim and W Bruce Croft. 2012. A field relevance model for structured document retrieval. In *Advances in Information Retrieval*, pages 97–108. Springer.
- Jinyoung Kim, Xiaobing Xue, and W Bruce Croft. 2009. A probabilistic retrieval model for semistructured data. In *Advances in Information Retrieval*, pages 228–239. Springer.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Xiao Li, Ye-Yi Wang, and Alex Acero. 2008. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM.
- Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 572–579. ACM.
- Xiao Li. 2010. Understanding the semantic structure of noun phrase queries. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1337–1345. Association for Computational Linguistics.
- Mehdi Manshadi and Xiao Li. 2009. Semantic tagging of web search queries. 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 861–869. Association for Computational Linguistics.
- Stelios Paparizos, Alexandros Ntoulas, John Shafer, and Rakesh Agrawal. 2009. Answering web queries using structured data sources. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 1127–1130. ACM.
- Nikos Sarkas, Stelios Paparizos, and Panayiotis Tsaparas. 2010. Structured annotations of web queries. In *Proceedings of the 2010 international conference on Management of data*, pages 771–782. ACM.
- Ioannis Tsochantaridis, 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. ACM.
- V Varpnik. 1995. The nature of statistical learning theory.

# Sentiment Classification for Movie Reviews in Chinese Using Parsing-based Methods

Wen-Juan Hou and Chuang-Ping Chang

Department of Computer Science and Information Engineering,  
National Taiwan Normal University,  
No.88, Sec. 4, Tingzhou Road, Wenshan Distinct, Taipei, Taiwan  
emilyhou@csie.ntnu.edu.tw; lukechang@std.ntnu.edu.tw

## Abstract

Sentiment classification is able to help people automatically analyze customers' opinions from the large corpus. In this paper, we collect some Chinese movie reviews from Bulletin Board System and aim at making sentiment classification so as to extract several frequent opinion words in some movie elements such as plots, actors/actresses, special effects, and so on. Moreover, we result in a general recommendation grade for users. Focusing on the movie reviews in Chinese, we propose a novel procedure which can extract the pairs of opinion words and feature words according to dependency grammar graphs. This parsing-based approach is more suitable for review articles with plenty of words. The grading results will be presented by a 5-grade scoring system. The experimental results show that the accuracy of our system, with the deviation of grades less than 1, is 70.72%, and the Mean Reciprocal Rank (MRR) value is 0.61. When we change the 5-grade scoring system into producing two values: one for recommendation and the other for non-recommendation, we get precision rates 71.23% and 55.88%, respectively. The result shows an exhilarating performance and indicates that our system can reach satisfied expectancy for movie recommendation.

## 1 Introduction

As the rapid growth of text data, text mining has been applied to discover hidden knowledge from text in many applications and domains. Nowadays, reviews are increasing with a rapid speed and are available over internet in natural languages. Sentiment analysis tries to identify and extract subject information from reviews. The problem of automatic sentiment analysis has received significant attention in recent years, largely due to the explosion of online social-oriented

content (e.g., user reviews, blogs, etc). Furthermore, sentiment analysis can be used in various ways and in many applications such as suggestion systems based on the user likes and ratings, recommendation systems, or insisting in election campaigns. As one of the important applications, sentiment classification targets to rate the polarity of a given text accurately towards a label or a score, predicting whether the expressive opinion in the text is positive, negative, or neutral.

Identifying the sentiment polarity is a complex task. To address the problem of sentiment classification, various methodologies have been applied earlier. Generally, there are two types of approaches tackling the sentiment classification task according to the knowledge the systems used. One is corpus-based and the other is lexicon-based. Corpus-based approaches are usually supervised, i.e., requiring training sets, and performing well when the training set is large enough and correctly labeled. The approaches are studied in (Bakliwal *et al.*, 2011; Bessalv *et al.*, 2011; Kennedy and Inkpen, 2006; Jin *et al.*, 2009; Narayanan *et al.*, 2009; Pang *et al.*, 2002; Schuller and Knaup, 2011). On the contrary, the lexicon-based approaches are mostly unsupervised, requiring a dictionary or a lexicon of pre-tagged words. Each word that is present in a text is compared against the dictionary. If a word is present in the dictionary, then the polarity value in the dictionary is added to the polarity score. The recent related works to lexicon-based approaches include (Baloglu and Aktas, 2010; Hu and Liu, 2004; Montejo-Raez *et al.*, 2012; Qiu *et al.*, 2009; Taboada *et al.*, 2011; Thet *et al.*, 2010; Zhao and Li, 2009). Additionally, some researchers use natural language processing techniques to discover statistical and/or linguistic patterns in the text in order to reveal the sentiment polarity. We can find such works in (Bonev

*et al.*, 2012; Harb *et al.*, 2008; Lin and He, 2009; Su *et al.*, 2008; Turney, 2002).

From the previous studies, we know the problem of sentiment analysis is of much attention by many researchers. Most of researchers investigate English reviews rather than ones with other languages. Therefore, we motivate to explore the movie reviews on Chinese Bulletin Board System (BBS). The aim of the study is to classify the sentiment of Chinese articles, and it will help readers understand the sentiment orientation. Besides, we are concerned with matching opinion with movie elements, called feature words in this paper. It is a finer-grained classification comparing to the article view.

The rest of this paper is organized as follows. Section 2 presents the overview of our system architecture. We describe the proposed method in details, i.e., the components of the system, in Section 3. The experimental data used by the system and the results achieved by the proposed methods are shown and discussed in Section 4. Finally, we express our main conclusions and the possible future directions.

## 2 Architecture Overview

Figure 1 shows the overall architecture of our methods for the sentiment analysis on movie reviews in the Chinese language. At first, we segment the words where the Chinese Knowledge Information Processing (CKIP) word segmentation system is utilized.<sup>1</sup> We then divide the document collection into two parts: one for training and one for testing. For the training corpus, we manually annotate opinion words to be positive, negative or neutral. In the following, we manually annotate feature words which are related to some movie elements such as plots, actors/actresses, special effects, and so on, and hence we get a list of feature words and their corresponding categories. After that, a Chinese parser is applied and we propose an algorithm to relate feature words to opinion words based on the parsing information. Subsequently, we apply an approach to determine the classification of opinion words and thus build an opinion word database. For the testing corpus, besides opinion and feature word lists, we use three more databases to help extract opinion and feature words. The rest processes are like ones of the training corpus, including parsing the documents and making an association between feature words

and opinion words. Finally, we calculate the scores of reviews and produce the movie recommendation results.

## 3 Methods

As shown in Figure 1, the methods of classifying sentiment are separated into several parts. The details of each part are explained in the following.

### 3.1 Word segmentation

We use CKIP word segmentation system in this phase. When we input a Chinese sentence, CKIP word segmentation system will segment the word and show the part of speech for each word. For example, if we input a Chinese sentence 我覺得很好看 wo-jue-de-hen-hao-kan ‘I thought it was very good to see’ to CKIP, the result will be “我(Nh) 覺得(Vt) 很(ADV) 好看(Vi).” It segments the sentence into four words 我 wo ‘I’, 覺得 jue-de ‘thought’, 很 hen ‘very’ and 好看 hao-kan ‘good to see’. The parts of speech are pronoun, transitive verb, adverb and intransitive verb for states, respectively. From the above example, we see a word in the Chinese language can be regarded as a “lexical item,” which is a sequence of one or more Chinese characters. In this paper, we use “word” to represent the “lexical item,” not limiting to the Chinese character numbers.

### 3.2 Opinion word manual annotation

In the Chinese language, the parts of speech of opinion words are subcategories of verbs, for example, Vi (intransitive verb for states), Vt (transitive verb for states or actions), and so on. There is no “adjective” tagged in Chinese, but the corresponding part of speech is “verb.” Therefore, we extract vocabularies that are verbs and treat them as opinion words. Meanwhile, we give the sentiment polarity as positive, negative and neutral for each opinion word. We also observe that adverbs can imply the strength of sentiment polarity. For example, 很 hen ‘very’ and 非常 fei-chang ‘very much’ can put emphasis on the opinion words. The negation words, 不 bu ‘no’ and 沒有 mei-you ‘not’, are able to put oppositeness on the opinion words. The parts of speech of above words are adverbs (ADV). Hence we extract the adverbs from the documents and annotate them as emphasis, oppositeness and irrelevance. Some examples of positive opinion words, negative opinion words, adverbs for emphasis and adverbs for oppositeness are listed in Table 1.

<sup>1</sup> <http://ckipsvr.iis.sinica.edu.tw/>

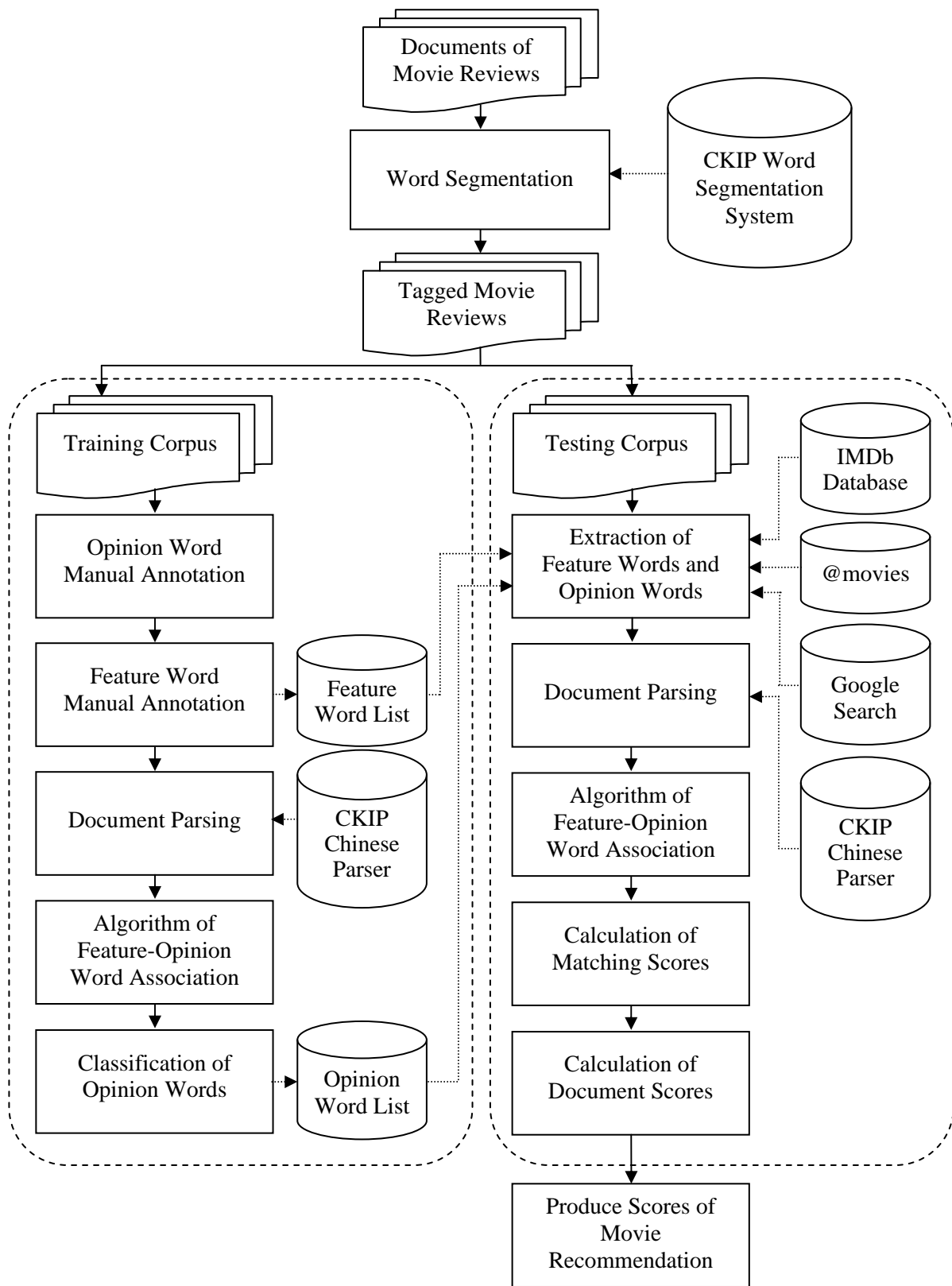


Figure 1. System architecture for the sentiment analysis on Chinese movie reviews.



Category	Examples
Positive opinion words	一氣呵成 yi-qi-he-cheng ‘ac-complish something at one go’, 鮮明 xian-ming ‘bright’, 討喜 tao-xi ‘satisfactory’
Negative opinion words	不清不楚 bu-qing-bu-chu ‘unclear’, 落伍 luo-wu ‘superannuated’, 莫名其妙 mo-ming-qi-miao ‘odd’
Adverbs for emphasis	十分 shi-fen ‘perfectly’, 愈來愈 yu-lai-yu ‘even more’, 格外 ge-wai ‘especially’
Adverbs for oppositeness	不可能 bu-ke-neng ‘impossible’, 尙未 shang-wei ‘not yet’, 無法 wu-fa ‘unable’

Table 1. Examples of opinion word annotations.

### 3.3 Feature word manual annotation

At this phase, we first manually annotate vocabularies related to movies from the training corpus and then build our own general feature word list. From the training corpus, we do not include some special proper noun related to movies such as names of actors/actresses (e.g., 湯姆克魯斯 tang-mu-ke-lu-si ‘Tom Cruise’) and character names (e.g., 哈利波特 ha-li-po-te ‘Harry Potter’) because it is not complete enough to cover all of the latest movies.

To include more complete movie-related people names, we reference to IMDb<sup>2</sup> and @movies<sup>3</sup> from which we get Chinese and English names of directors, playwrights and stars. Because authors often use hypocorisms or different translated Chinese names in the review articles, we search for hypocorisms and translated Chinese names from Google.<sup>4</sup> Finally, we combine searching data from Google with information from IMDb and @movies, and build a specific feature word list.

For classifying sentiment words at a finer-grained level, we reference to the study of Zhuang *et al.* (2006) and give the classification to each feature word. In this paper, we design four categories, including (1) entirety of movie, (2) story, (3) people (directors, playwrights, actors/actresses and characters), and (4) special effect and others. Table 2 shows some annotated feature word examples for each category.

<sup>2</sup> <http://www.imdb.com>

<sup>3</sup> <http://www.atmovies.com.tw>

<sup>4</sup> <http://www.google.com.tw>

Category	Examples
Entirety of movie	電影 dian-ying ‘movie’, 影片 ying-pian ‘film’, 場面 chang-mian ‘scene’
Story	伏筆 fu-bi ‘foreshadow’, 劇本 ju-ben ‘scenario’, 情節 qing-jie ‘action’
People	主角 zhu-jiao ‘leading role’, 人物 ren-wu ‘figure’, 配角 pei-jiao ‘minor actor’
Special effect and others	主題曲 zhu-ti-qu ‘theme song’, 動畫 dong-hua ‘animation’, 布景 bu-jing ‘stage settings’

Table 2. Examples of feature word annotations.

### 3.4 Document parsing

The structure of a sentence is important for understanding and analysis of sentence semantics. In this phase, we utilize CKIP Chinese parser for the parsing task.<sup>5</sup> The parser is based on probabilistic context-free grammar and is refined with probabilities of word-to-word association in disambiguation. After parsing, we get the syntactic structure trees of the documents in the corpus.

### 3.5 Algorithm of feature-opinion word association

In a sentence of movie reviews, some feature word is associated with some opinion word. Generally, most of algorithms (e.g., Hu and Liu, 2004) relate a feature word to an opinion word if they collocate closely in a sentence. For example, a sentence 這部電影很吸引人 zhe-bu-dian-ying-hen-xi-yin-ren ‘The movie is very attractive’ includes an association between the feature word (電影 dian-ying ‘movie’) and the opinion word (吸引人 xi-yin-ren ‘attractive’). But the algorithm is too simple to find all correct associations. Let us look at the following sentence.

再好看的電影也會變得無聊 zai-hao-kan-de-dian-ying-ye-hui-bian-de-wu-liao ‘The even more interesting film will become boring too.’ (1)

If we use a simple rule of only considering collocation, the opinion word 好看 hao-kan ‘interesting’ will associate with the feature word 電影 dian-ying ‘film’. But the semantics of the sentence means there should be an association be-

<sup>5</sup> <http://godel.iis.sinica.edu.tw/CKIP/parser.htm>

tween the feature word 電影 dian-ying ‘film’ and the opinion word 無聊 wu-liao ‘boring’. It is due to no syntactic information is adapted. Consequently, we introduce a Chinese parser and try to solve this problem.

From analyzing the parsing tree, we find the feature word 電影 dian-ying ‘film’ and the opinion word 無聊 wu-liao ‘boring’ are at the same level where the feature word 電影 dian-ying ‘film’ is related to the opinion word 無聊 wu-liao ‘boring’. It is an important clue in the study.

The other problem occurs if there are two feature words appearing in a sentence because we have to decide which feature word (or both words) should be related to the opinion word. Let us see the following sentences.

周杰倫的電影實在不吸引人 zhou-jie-lun-de-dian-ying-shi-zai-bu-xi-yin-ren ‘The film of Jay Chou is not attractive at all.’ (2)

周杰倫和電影都不吸引人 zhou-jie-lun-han-dian-ying-dou-bu-xi-yin-ren ‘Jay Chou and his film are not attractive at all.’ (3)

In Sentence (2), the opinion word 不吸引人 bu-xi-yin-ren ‘not attractive at all’ is to modify the feature word 電影 dian-ying ‘film’. But in Sentence (3), the opinion word 不吸引人 bu-xi-yin-ren ‘not attractive at all’ is to modify the feature words 周杰倫 zhou-jie-lun ‘Jay Chou’ and 電影 dian-ying ‘film’. We investigate this problem using the information of the syntactic structure tree.

By the above analysis, we propose the following algorithm to make an association between feature words and opinion words.

1. Traverse the syntactic structure tree by breadth-first search, and get the levels for all nodes.
2. Starting from the root, find whether there exists any feature word or opinion word.
  - 2.1 If feature words and opinion words are found, associate each feature word to all opinion words. For example, if there are three feature words and two opinion words, there will be six pairs of association. Stop searching in the tree.
  - 2.2 If only feature words exist, search for a subtree rooted with VP (verb phrase). If there is, search all nodes in the VP subtree for opinion words. If there is no VP subtree or no opinion word exists, stop searching in the tree.
  - 2.3 If only opinion words exist, search for a subtree rooted with NP. If there is,

search all nodes in the NP subtree for feature words. If there is no NP subtree or no feature words, stop searching in the tree.

- 2.4 If there are no opinion words and feature words, but we can find a subtree rooted with NP and a subtree rooted with VP, then search for feature words in the NP subtree and opinion words in the VP subtree. Make associations with all feature words and opinion words.
- 2.5 If no association is found in Steps 2.1 – 2.4, recursively, search the subtree at the different level and repeat Step 2.
3. At the levels of existing feature words and opinion words, find if there is any adverb built in our list. If there is, add the adverb to the feature-opinion word pair.
4. When the system stops searching for feature words and opinion words, only opinion words are extracted from the tree. Associate the opinion words with the special feature word NULL.
5. According to the category of feature words, the pair of feature-opinion words is put into the proper category. The special feature word NULL is classified to a new category.

Applying the algorithm, the matching pairs of Sentences (1) – (3) are presented in Table 3.

Sentence	Matching Pair
(1)	電影 dian-ying ‘film’ – 無聊 wu-liao ‘boring’
(2)	電影 dian-ying ‘film’ – 不 bu ‘not’ – 吸引人 xi-yin-ren ‘attractive’
(3)	周杰倫 zhou-jie-lun ‘Jay Chou’ – 不 bu ‘not’ – 吸引人 xi-yin-ren ‘attractive’ 電影 dian-ying ‘film’ – 不 bu ‘not’ – 吸引人 xi-yin-ren ‘attractive’

Table 3. Matching pairs for Sentences (1), (2) and (3).

### 3.6 Classification of opinion words

Based on the matching pairs extracting from Section 3.5, we can count the number of opinion words in the different categories. We introduce the concept of Term Frequency – Inverse Document Frequency (TF – IDF) to compute the importance of opinion words in the specific categories. The equations are listed as follows.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

$$idf_{i,j}^* = \log \frac{\sum_p n_{i,p}}{1 + \sum_{p,p \neq j} n_{i,p}} \quad (2)$$

$$tf\_idf_{i,j}^* = tf_{i,j} \times idf_{i,j}^* \quad (3)$$

where  $n_{i,j}$  is the number of opinion word  $t_i$  appearing in the category  $g_j$ .  $tf_{i,j}$  is the normalized frequency of term  $t_i$  appearing in the category  $g_j$ .  $idf_{i,j}^*$  is a variation of traditional  $idf_{i,j}$ . In the traditional  $idf_{i,j}$ , it is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient. In this study, there are only four categories, the numerator will be 4 and it causes no discrimination with other terms. Hence, we design a new  $idf_{i,j}^*$  as Equation (2). The numerator is the total number of opinion word  $t_i$  in all categories. The denominator is one plus the total count of opinion word  $t_i$  in other three categories that  $t_i$  does not belong to (a summand one is for avoiding divided by zero).

According to our previous approach, the category of opinion words is the same as the associated feature word. If there is no feature word in the sentence but with an opinion word in a sentence, we must devise some strategy to decide the category. To solve this problem, we propose two thresholds  $M_1$  and  $M_2$ , and Equation (4). In this way, all opinion words  $t_i$  can have their mapping categories  $g_j$ .

$$t_i \in g_i \text{ if } tf\_idf_{i,j}^* > M_1 \text{ and } \frac{tf_{i,j} - idf_{i,j}^*}{\max\{tf\_idf_{i,k} \mid \forall k, k \neq j\}} > M_2 \quad (4)$$

We assign  $M_1=2.0$  and  $M_2=0.02$ . Table 4 lists examples of opinion words using Equation (4).

Category	Opinion Word Examples
Entirety of movie	推薦 tui-jian ‘recommend’ 叫座 jiao-zuo ‘box-office’
Story	動人 dong-ren ‘touching’ 冗長 rong-chang ‘long’
People	迷人 mi-ren ‘charming’ 成熟 cheng-shou ‘mature’
Special effect and others	逼真 bi-zhen ‘lifelike’ 震撼 zhen-han ‘vibrate’

Table 4. Examples of opinion words.

### 3.7 Calculation of document scores

Now, we have a set of matching pairs between opinion words and feature words. We design an

equation to compute the scores for the pairs. The equation is listed in Equation (5).

$$S_{i,j} = opS(Opinion_{i,j}) \times \prod_k advS(Adv_{i,j,k}) \quad (5)$$

where  $S_{i,j}$  presents the score of the  $i$ th matching pair in category  $g_j$ .  $Opinion_{i,j}$  stands for the opinion word of the  $i$ th matching pair in category  $g_j$ . The function  $opS$  will output 1 if  $Opinion_{i,j}$  is a positive sentiment vocabulary, and output  $-1$  if  $Opinion_{i,j}$  is a negative sentiment vocabulary.  $Adv_{i,j,k}$  is the adverb of the  $i$ th matching pair in category  $g_j$ . Since there is perhaps more than one adverb in the matching pair, we give a third index  $k$  in the equation. The function  $advS$  has values of 1.2 and  $-1$  as its range. If it produces the value 1.2, it means the adverb  $adv_{i,j,k}$  is used for emphasis on the opinion word  $Opinion_{i,j}$ . If it generates the value  $-1$ , it means the adverb  $adv_{i,j,k}$  puts oppositeness on the opinion word  $Opinion_{i,j}$ . The final score of the  $i$ th matching pair in category  $g_j$  is the product of  $opS$  and  $advS$ .

When we get the scores of all opinion words in the categories, we can compute the score of the opinion word in the review article that is the summation of individual scores. The equation is listed as below.

$$Score = \sum_{j=1}^4 \sum_k S_{k,j} \quad (6)$$

where  $Score$  is the sentiment score of the document. We then map  $Score$  into five levels, i.e., 1, 2, 3, 4 and 5, and call it as “level score”. Levels 1 and 2 identify that the document is with negative sentiment, and level 1 is more negative than level 2. Level 3 means that the document is neutral. Levels 4 and 5 identify the positive sentiment polarity in the document, and level 5 is stronger.

## 3.8 Make movie recommendation

In the last phase, we average the level scores of reviews for each movie, and then the recommendation for each movie is presented with the averaged level scores.

## 4 Experiments and Results

### 4.1 Experimental data

The experimental data were obtained from the movie discussion board of website ptt BBS.<sup>6</sup> Users can post their opinions on BBS that acts as a platform for users to share their opinions. From the latest movies, we first select seven movies

<sup>6</sup> telnet://ptt.cc

with different styles. The movies are Mission: Impossible - Ghost Protocol, Treasure Hunter, The A-Team, The Avengers, Toy Story 3, You are the Apple of My Eye, and The Hunger Games. Then we automatically retrieve 50 articles for each movie. To get richer information, the length of retrieved articles is restricted to more than 100 Chinese words. In the following, we filter out the articles that are irrelevant to reviews. For example, some articles are filled with the same words such as 好看 hao-kan ‘good to see’ and no other words are appeared, so the articles will be filtered out. Finally, we get 321 articles with 379,360 Chinese words.

## 4.2 Experimental results and discussion

In our experiments, we retrieve 11,837 pairs of feature-opinion word matching where 6,906 pairs are useful and 4,931 pairs are useless. The 6,906 pairs are used as evaluation.

**Evaluation of reliability of agreement:** First, we invite three humans (A, B and C) who often go to the movies to read the review articles and give the level scores (1, 2, 3, 4, and 5). The average scores will be the gold standard for evaluation. For assessing the reliability of agreement between three scores the humans give, we use the weighted Kappa coefficient (Sim and Wright, 2005) with the quadratic weighting scheme. The formula is given as below.

$$k_w = \frac{\sum (w \cdot f_o) - w \cdot f_c}{n - \sum (w \cdot f_c)},$$

$$quadratic\ weight = 1 - \left(\frac{i-j}{k-1}\right)^2 \quad (7)$$

where  $w$  is the weight,  $f_o$  is the value of the observed disagreement, and  $f_c$  is the value of the chance disagreement.  $k$  is the number of the ratings and  $(i-j)$  is the value of disagreement. Kappa’s possible values are constrained to the interval  $[0, 1]$ ;  $k_w=0$  means that agreement is not different from by chance, and  $k_w=1$  means perfect agreement.

The agreement evaluation results are shown in Table 5. It shows that human answers agree with each other almost perfect since the values of the weighted kappa are larger than 0.8.

Human Pairs	Weighted Kappa $k_w$
A, B	0.87
A, C	0.89
B, C	0.89

Table 5. Human agreement evaluation results.

### Evaluation of results with human scores:

We use the Mean Reciprocal Rank (MRR) to evaluate the difference between the scores produced by the system and the scores given by the humans. The formula of MRR is as follows.

$$MRR = \frac{1}{|A|} \sum_{i=1}^{|A|} \frac{1}{rank_i}, \quad (8)$$

$$rank_i = |human(A_i) - system(A_i)| + 1$$

where  $A$  is the set of all review documents.  $|A|$  is the number of review documents.  $A_i$  is the  $i$ th review document.  $human(A_i)$  is the average score given by humans.  $system(A_i)$  is the score given by our proposed system.  $rank_i$  means the difference between humans and the system, and a summand 1 is used for avoiding divided by zero. The MRR value to our system is 0.61. Table 6 shows the rank distribution.

$rank_i$	Article Numbers	Ratio
1	110	34.27%
2	117	36.45%
3	67	20.87%
4	18	5.61%
5	9	2.80%

Table 6. Rank distribution of the evaluation.

From Table 6, 34.27% articles get the same scores between the system and humans. If the score difference of the system and humans is less than 1, there are 70.72% articles. Only 8.41% articles have the deviation greater than 2. The result demonstrates that the scores produced by our system are reliable.

If we classify the scores of 4 and 5 as recommendable and others as non-recommendable, then the evaluation results are shown in Table 7.

Class	Totals	System	Correct	Recall	Precision
Re	201	219	156	77.61%	71.23%
Non-re	120	102	57	47.50%	55.88%

Table 7. Rank distribution of the evaluation.

In Table 7, ‘‘Re’’ means recommendable and ‘‘Non-re’’ means non-recommendable. ‘‘Totals’’ is the article amount that humans give and ‘‘System’’ is the article amount that the system produces. ‘‘Correct’’ presents number of consistency between humans and the system. ‘‘Recall’’ is the value of ‘‘Correct’’ dividing ‘‘Totals’’. ‘‘Precision’’ is the value of ‘‘Correct’’ dividing ‘‘System’’. The result shows that the performance of ‘‘recommendable’’ is better than ‘‘non-recommendable’’.

To explore the difference between humans and the system more detailed, we observe that most of the scores generated by the system and humans are similar. Especially, there is no difference for the movie “Mission: Impossible”. The largest difference exists in the movie “Treasure Hunter”. It is because the reviews have some sentences damn the movie with faint praise, i.e., with negative sentiment but seems positive to the system. Let us see the following sentence.

很意外的發現這是一部中等以上的優良惡搞片 hen-yi-wai-de-fa-xian-zhe-shi-yi-bu-zhong-deng-yi-shang-de-you-liang-e-gao-pian ‘Unexpectedly we find that it is a good spoof movie with an average level or better.’ (4)

For Sentence (4), the system produces a feature-opinion pair of “movie – good” and marks it positive. But in effect, the opinion should be negative. It is the main reason why the performance of non-recommendation is worse.

In addition, the system can retrieve opinion words that reviewers often use for different categories. Table 8 lists some opinion words that are frequently used for the movie “Mission: Impossible - Ghost Protocol”. These extracted opinion words are reasonable for commenting about a Hollywood action movie.

Category	Opinion Words
Story	緊湊 jin-cou ‘compact’ 刺激 ci-ji ‘exciting’ 幽默 you-mo ‘humorous’ 驚人 jing-ren ‘amazing’
People	好 hao ‘good’ 強 qiang ‘strong’ 帥氣 shuai-qi ‘smart’ 鮮明 xian-ming ‘bright’
Special effect and others	經典 jing-dian ‘classic’ 精彩 jing-cai ‘splendid’ 罕見 han-jian ‘rarely seen’ 豐富 feng-fu ‘rich’

Table 8. Some frequently used opinion words.

#### Evaluation of results with IMDb scores:

Except for the scores produced by humans, we also compare the results with the scores in the IMDb website. IMDb is a popular movie database and its registered members can give scores for movies. This aspect of evaluation will tell us the consistence between Taiwanese and members in the IMDb website. Because the scores pro-

posed by IMDb are ranged from 1 to 10, we divide them by 2 to mapping into our 5-graded scores. The result is shown in Table 9.

Movies	System	IMDb
Mission: Impossible - Ghost Protocol	4.1	3.7
Treasure Hunter	3.5	2.0
The A-Team	4.0	3.5
The Avengers	4.1	4.3
Toy Story 3	4.2	4.3
You are the Apple of My Eye	3.9	3.8
The Hunger Games	2.9	3.8

Table 9. Comparison to the system and IMDb.

Although the reviewers from IMDb are different from ones from our movie discussion board, the comparison also demonstrates that the recommendable trend is consistent.

## 5 Conclusion

In this paper, we propose a sentiment classification system based on parsing models. We present a matching algorithm for feature-opinion words, and make effective analysis for reviews with plenty of words.

The experimental results show 70.72% precision rate under the difference less than 1. If the scores are mapped to two levels (recommendable, non-recommendable), the precision rates are 71.23% and 55.88%, respectively. We also compare the result with a popular movie website IMDb, and we discover most of the score trend is similar. It shows the results are exhilarating and indicates that our system can reach satisfied expectancy for movie recommendation.

In the future, we plan to adapt learning methods for matching feature words and opinion words. Besides, we want to explore word polarity according to opinion holders. It will help users understand sentiment orientation for each review more thoroughly.

#### Acknowledgments

Research of this paper was partially supported by National Science Council, Taiwan, under the contract NSC 102-2221-E-003-027.

#### References

- Akshat Bakliwal, Piyush Arora, Ankit Patil and Vasudeva Varma. 2011. Towards Enhanced Opinion Classification Using NLP Techniques. *Proceedings of the Workshop on Sentiment Analysis where AI Meets Psychology, IJCNLP 2011*, 101 – 107, Chiang Mai, Thailand, November 13, 2011.

- Arzu Baloglu and Mehmet S. Aktas. 2010. BlogMiner: Web Blog Mining Application for Classification of Movie Reviews. *Proceedings of 2010 5<sup>th</sup> International Conference on Internet and Web Applications and Services*, 77 – 84, Barcelona, Spain, May 5 – 19, 2010.
- Dmitriy Beshpalov, Bing Bai, Yanjun Qi and Ali Shokoufandeh. 2011. Sentiment Classification Based on Supervised Latent n-gram Analysis. *Proceedings of the 20<sup>th</sup> ACM Conference on Information and Knowledge Management*, 375 – 382, Glasgow, UK, October 24 – 28, 2011.
- Boyan Bonev, Gema Ramirez-Sanchez and Sergio Ortiz Rojas. 2012. Opinum: Statistical Sentiment Analysis for Opinion Classification. *Proceedings of the 3<sup>rd</sup> Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, 29 – 37, Jeju, Republic of Korea, July 12, 2012.
- CKIP word segmentation system. Available at <http://ckipsvr.iis.sinica.edu.tw/>.
- Ali Harb, Michel Plantié, Gerard Dray, Mathieu Roche, François Troussel and Pascal Poncelet. 2008. Web Opinion Mining: how to Extract Opinions from Blogs? *Proceedings of the 5<sup>th</sup> International Conference on Soft Computing as Transdisciplinary Science and Technology*, Cergy-Pontoise, France, October 27 – 31, 2008.
- Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168 – 177, Seattle, WA, USA, August 22–25, 2004.
- Wei Jin, Hung Hay Ho and Rohini K. Srihari. 2009. OpinionMiner: A Novel Machine Learning System for Web Opinion Mining and Extraction. *Proceedings of the 15<sup>th</sup> ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1195 – 1204, Paris, France, Jun 28 – July 1, 2009.
- Alistair Kennedy and Diana Inkpen. 2006. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22 (2): 110 – 125.
- Chenghua Lin and Yulan He. 2009. Joint Sentiment/Topic Model for Sentiment Analysis. *Proceedings of the 18<sup>th</sup> ACM Conference on Information and Knowledge Management*, 375 – 384, Hong Kong, China, November 2 – 6, 2009.
- A. Montejo-Raez, E. Martinez-Camara, M. T. Martin-Valdivia, L. A. Urena-Lopez. 2012. Random Walk Weighting over SentiWordNet for Sentiment Polarity Detection on Twitter. *Proceedings of the 3<sup>rd</sup> Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, 3 – 10, Jeju, Republic of Korea, August 12, 2012.
- Ramanathan Narayanan, Bing Liu and Alok Choudhary. 2009. Sentiment Analysis of Conditional Sentences. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 180 – 189, Singapore, August 6 – 7, 2009.
- Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 79 – 86, University of Pennsylvania, Philadelphia, PA, USA, July 6 – 7, 2002.
- Likun Qiu, Weishi Zhang, Changjian Hu and Kai Zhao. 2009. SELC: A Self-Supervised Model for Sentiment Classification. *Proceedings of the 18<sup>th</sup> ACM Conference on Information and Knowledge Management*, 929 – 936, Hong Kong, China, November 2 – 6, 2009.
- Bjorn Schuller and Tobias Knaup. 2011. Learning and Knowledge-based Sentiment Analysis in Movie Review Key Excerpts. *Proceedings of the 3<sup>rd</sup> COST 2102 International Training School*, 448 – 472, Caserta, Italy, March 15–19, 2011.
- Julius Sim and Chris C. Wright. 2005. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85:257–268.
- Qi Su, Xinying Xu, Honglei Guo, Zhili Guo, XianWu, Xiaoxun Zhang, Bin Swen and Zhong Su. 2008. Hidden Sentiment Association in Chinese Web Opinion Mining. *Proceedings of the 7<sup>th</sup> International World Wide Web Conference*, 959 – 968, Beijing, China, April 21 – 25, 2008.
- Maite Taboata, Julian Brooke, Milan Tofiloski, Kimberly Voll and Manfred Stede. 2011. Lexicon-based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267 – 307.
- Tun Thura Thet, Jin-Cheon Na and Christopher S.G. Khoo. 2010. Aspect-based Sentiment Analysis of Movie Reviews on Discussion Boards. *Journal of Information Science*, 36 (6): 823 – 848.
- Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40<sup>th</sup> Annual Meeting of the Association for Computational Linguistics (ACL)*, 417 – 424, July 6 – 12, 2002, Philadelphia, PA, USA.
- Lili Zhao and Chunping Li. 2009. Ontology Based Opinion Mining for Movie Reviews. *Proceedings of the 3<sup>rd</sup> International Conference on Knowledge Science, Engineering and Management*, 204 – 214, University of Vienna, Austria, November 25 – 27, 2009.

# Sentiment Aggregation using ConceptNet Ontology

Subhabrata Mukherjee and Sachindra Joshi

IBM India Research Lab

{subhabmu, jsachind}@in.ibm.com

## Abstract

Sentiment analysis of reviews traditionally ignored the association between the features of the given product domain. The hierarchical relationship between the features of a product and their associated sentiment that influence the polarity of a review is not dealt with very well. In this work, we analyze the influence of the hierarchical relationship between the product attributes and their sentiments on the overall review polarity. ConceptNet is used to automatically create a product specific ontology that depicts the hierarchical relationship between the product attributes. The ontology tree is annotated with feature-specific polarities which are aggregated bottom-up, exploiting the ontological information, to find the overall review polarity. We propose a weakly supervised system that achieves a reasonable performance improvement over the baseline without requiring any tagged training data.

## 1 Introduction

In recent years there has been a huge surge of activity in the social networking sites, blogs and review sites. The voluminous amount of data generated is a goldmine of information for the retail brands to find out the customer needs, concerns and potential market segments. Sentiment analysis aims to mine this information to find out the popular sentiment about any product and its associated features.

Traditionally sentiment analysis has been posed as a text classification task on features derived from the given text. In the product review domain, the initial works in sentiment analysis focused on classifying the entire review as positive or negative using various word-based and phrase-based features (Turney *et al.*, 2003; Turney 2002; Kamps *et al.*, 2002; Hatzivassiloglou *et al.*, 2000; Hatzivassiloglou *et al.*, 2002). The more recent works focused on product feature

extraction from a review and performing feature-specific sentiment analysis (Hu *et al.*, 2004; Mukherjee *et al.*, 2012). For example, the review, *The audio quality of my new phone is absolutely awesome but the picture taken by the camera is a bit grainy*, is positive with respect to the *audio quality* and negative with respect to the *camera*. However, once the feature-specific polarities are obtained, the works do not describe any systematic approach to aggregate the feature-specific polarities to obtain the overall review polarity. A naïve count-based feature-specific polarity aggregation will not work well for reviews having different features with diverse opinions. A bag-of-words based model will pick up *awesome* and *grainy* as the sentiment features and mark the overall review as *neutral*. One may argue that the *audio quality* is more important to a *cell phone* than the *camera* and hence the overall review polarity should be positive. While the feature-specific model associates sentiment to features, it cannot do a polarity aggregation in absence of feature association information to find the overall review polarity.

Let us consider the following review taken from Amazon.com which more clearly depicts the necessity of learning the hierarchical product-attribute relationship and associated sentiments.

*I bought a Canon EOS 7D (DSLR). It's very small, sturdy, and constructed well. The handling is quite nice with a powder-coated metal frame. It powers on quickly and the menus are fairly easy to navigate. The video modes are nice, too. It works great with my 8GB Eye-Fi SD card. A new camera isn't worth it if it doesn't exceed the picture quality of my old 5Mpixel SD400 and this one doesn't. The auto white balance is poor. I'd need to properly balance every picture taken so far with the ELPH 300. With 12 Mpixels, you'd expect pretty good images, but the problem is that the ELPH 300 compression is turned up so high that the sensor's acuity gets lost (softened) in compression.*

The above example depicts the complexity involved in analyzing product reviews. The review has a mix of good and bad comments about various features of the product. A flat classification model which considers all features to be equally important will fail to capture the proper polarity of the review. The reviewer seems happy with the *camera size, structure, easy use, video modes, SDHC support etc.* However, the *auto-white balance* and *high compression* leading to sensor acuity seem to disappoint him. Now, the primary function of a camera is to take good pictures and videos. Thus *picture, video quality, resolution, color balance etc.* are of primary importance whereas *size, video mode, easy use etc.*, are secondary in nature. The overall review polarity should be negative as the reviewer shows concerns about the most important features of the camera.

In this paper, we propose a weakly supervised approach to aggregate the sentiment about various features of a product to give the overall polarity of the review, without requiring expensive labeled training data. The approach is weakly supervised due to the requirement of ConceptNet (created by crowd-sourcing), a dependency parser and a sentiment lexicon.

The objectives of the paper can be summarized as:

1. Automatically learning the product-attribute hierarchy from a knowledge resource, where we leverage ConceptNet (Hugo *et al.*, 2004) to learn the product *attributes, synonyms, essential components, functionalities etc.* and create a domain specific ontology tree
2. Discovering the various features of a product in the review and extracting feature-specific sentiment
3. Mapping the product features with their associated sentiments to the ontology tree and aggregating the feature-specific sentiments to determine the overall review polarity

## 2 Related Works

The initial works in sentiment analysis used bag-of-words features like unigrams, bigrams, adjectives *etc.* which gave way to the usage of phrase-based features like part-of-speech sequences (Ex: adjectives followed by nouns) (Turney *et al.*, 2003; Turney 2002; Kamps *et al.*, 2002; Hatzivassiloglou *et al.*, 2000; Hatzivassiloglou *et al.*, 2002). These works did not consider the attributes or features of the underlying product domain in the review. A review may contain multiple

features with a different opinion about each feature. This makes it difficult to come up with an overall polarity of the review. The latter works addressed this issue by focusing on feature-specific sentiment analysis.

Feature-specific sentiment analysis attempts to find the polarity of a review with respect to a given feature. Approaches like dependency parsing (Wu *et al.*, 2009; Chen *et al.*, 2010; Mukherjee *et al.*, 2012), joint sentiment topic model using LDA (Lin *et al.*, 2009) have been used to extract feature-specific expressions of opinion. Although these works extract the feature-specific polarities, they do not give any systematic approach to aggregate the polarities to obtain the overall review polarity.

Wei *et al.* (2010) propose a hierarchical learning method to label a product's attributes and their associated sentiments in product reviews using a Sentiment Ontology Tree (HL SOT). Although our work stems from a similar idea, it differs in a number of ways. The HL-SOT approach is completely supervised, requiring the reviews to be annotated with *product-attribute relations*, as well as *feature-specific opinion expressions*. The approach requires a lot of labeling information which needs to be provided for every domain. Also, the authors do not describe any elegant approach to aggregate the feature-specific polarities of the children nodes to obtain the overall review polarity.

In this work, we use ConceptNet (Hugo *et al.*, 2004) as a knowledge resource to automatically construct a domain-specific ontology tree for product reviews, without requiring any labeled training data. ConceptNet relations have an inherent structure which helps in the construction of an ontology tree from the resource. ConceptNet has been used in information retrieval tasks in other domains (Guadarrama *et al.*, 2008; Kotov *et al.*, 2012). But there has been a very few works (Sureka *et al.*, 2010) in sentiment analysis using ConceptNet. Unlike the previous works, we present an approach to deal with noisy and one-to-many relations in ConceptNet as well as the myriad of relations and the ensuing topic drift. We also present a novel sentiment aggregation approach to combine the feature-specific polarities with ontological information to find the overall polarity of the review.

## 3 Ontology Creation from ConceptNet

Ontology can be viewed as a knowledge base, consisting of a structured list of concepts, rela-



tions and individuals (Estival *et al.*, 2004). The hierarchical relationship between the product attributes can be best captured by an *Ontology Tree*. Wei *et al.* (2010) use a tree-like ontology structure that represents the relationships between a product’s *attributes* or *features*. They define a Sentiment Ontology Tree (SOT) where each of the non-leaf nodes of the SOT represents an attribute of a camera and all leaf nodes of the SOT represent sentiment (positive/negative) nodes respectively associated with their parent nodes.

We adopt a similar idea and consider an *Ontology Tree* for a product domain (say, *camera*) where the *feature nodes* (attributes like *body*, *lens*, *flash* etc.) are annotated with *feature-specific polarities* of the review.

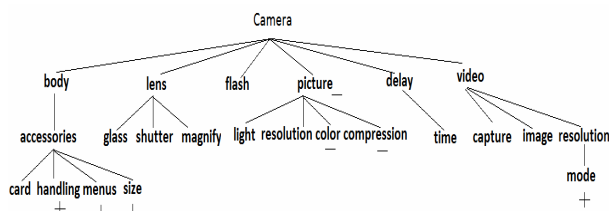


Figure 1. Snapshot of Camera Ontology Tree

The *feature nodes* in our *ontology tree* depict features of interest or attributes (Ex: *lens*, *flash*, *picture* etc.) of the given product (Ex: *camera*). The edges in the ontology tree depict the *relation type* connecting a feature with its parent. For example, a *lens* is a *partof* a *camera*, a *camera* is *usedfor* taking *pictures*, *time\_delay* is *derivedfrom* *time* etc. The *feature nodes* are annotated with *polarities* (+ and – denoting positive and negative sentiment, respectively) of the feature with respect to the review.

Figure 1 shows a snapshot of the ontology tree of a camera for the given example review in Section 1. The figure shows more positive feature-polarities than negative feature-polarities, but the review is still negative. This is because the feature polarities in the higher level of the ontology tree dominate those at a lower level, *i.e.* the importance of a feature dilutes with the increase in the ontology depth.

### 3.1 Domain Ontology Tree Creation

In this work, we leverage ConceptNet (Hugo *et al.*, 2004) to construct a domain-specific ontology tree for product reviews. ConceptNet is a very large semantic network of common sense knowledge which can be used to make various inferences from text. It is the largest, machine-usable common sense resource consisting of more than 250,000 propositions. Mining infor-

mation from ConceptNet can be difficult as one-to-many relations, noisy data and redundancy undermine its performance for applications requiring higher accuracy (Smith *et al.*, 2004). However, we use ConceptNet for the following reasons:

1. The relational predicates in ConceptNet have an inherent structure suitable for building ontology. For example, relations like *partof*, *hasa*, *madeof* can be readily conceptualized as hierarchical relations.
2. ConceptNet has a closed class of well-defined relations. The relations can be suitably weighted and used for various purposes.
3. The continual expansion of the knowledge resource through crowd-sourcing incorporates new data and enriches the ontology.
4. Ontology creation using ConceptNet does not require any labeling of product reviews.

#### 3.1.1 ConceptNet Relations

ConceptNet<sup>1</sup> has a closed class of 24 primary relations, expressing connections between various concepts.

camera	UsedFor	take_picture
camera	IsA	tool_for_take_picture
camera	AtLocation	store
tripod	UsedFor	keep_camera_steady
camera	CapableOf	record_image
camera	IsA	device
flash	PartOf	camera
lens	AtLocation	camera
tripod	AtLocation	camera_shop
camera	IsA	photo_device
cannon	ConceptuallyRelatedTo	camera
photograph	ConceptuallyRelatedTo	camera
picture	ConceptuallyRelatedTo	camera

Table 1. ConceptNet Relation Examples

We categorize the ConceptNet relations into 3 primary categories – *hierarchical* relations, *synonymous* relations and *functional* relations. *Hierarchical* relations represent parent-child relations and can be used to construct the tree top-down, as the relations are transitive. *Synonymous* relations help to identify related concepts. Thus similar nodes can be merged during tree construction. *Functional* relations help to identify the purpose or property of interest of the concept. The relation categorization helps to weigh various relations differently. Consider the case where the *functional* relation “a camera is *usedfor* taking\_picture” may be of more interest to an individual than the hierarchical relation “a camera

<sup>1</sup><http://csc.media.mit.edu/docs/conceptnet/conceptnet4.html#relations>

hasa tripod”. Thus a product which takes good pictures but lacks a tripod will have a high positive polarity. This is, of course, subjective and can be used to personalize the ontology tree. The other advantage of relation categorization is to deal with one-to-many relations, as will be discussed in the next section.

Hierarchical	: LocatedNear, HasA, PartOf, MadeOf, IsA, InheritsFrom
Synonymous	: Synonym, ConceptuallyRelatedTo
Functional	: UsedFor, CapableOf, HasProperty, DefinedAs

**Table 2.** ConceptNet Relation Type Categorization

### 3.1.2 Algorithm for Ontology Construction

Ontology construction from ConceptNet is hindered by the following obstacles:

1. One-to-many relations exist between the concepts. For example, the concepts *camera* and *picture* can be associated by relations like - camera *UsedFor* take\_picture, camera *HasA* picture, picture *ConceptuallyRelatedTo* camera, picture *AtLocation* camera etc.

2. There is a high degree of *topic drift* during relation extraction. For example, the predicates camera *HasA* lens, lens *IsA* glass and glass *HasA* water places water at a high level in the ontology tree, although it is not at all related to camera.

The hierarchical relations in ConceptNet are much more definitive, have much less topic drift and can be used to ground the ontology tree. Hence, they are preferred over other relations during a relational conflict. In the above example, where *picture* is *ConceptuallyRelatedTo* *camera*, putting camera and picture at the same level will generate an incorrect ontology tree. The issue can be averted by preferring the hierarchical relation between camera and picture over the synonymous relation. The relational conflict is averted by ordering the predicate relations where hierarchical relations > synonymous relations > functional relations. In order to avoid topic drift, the ontology feature nodes extracted from ConceptNet are constrained to belong to a list of frequently found concepts in the domain, which is obtained from an unlabeled corpus.

In the first step of ontology construction, all the unlabeled reviews in the corpus are Part-of-Speech tagged and all *Nouns* are retrieved. The frequently occurring concepts are then added to the *feature set*. In the second step, the ConceptNet relations are partitioned into three disjoint sets *hierarchical*, *synonymous* and *functional*. The domain name is taken as the root of the *Ontology Tree*.

Input: Raw unlabeled corpus of product reviews and ConceptNet Knowledge Network

1. Part-of-speech tag the reviews and retrieve all *Nouns*. Let  $\bar{N}$  be the set of all potential features.
  2. A feature  $n_i \in \bar{N}$  is considered relevant and added to the feature set  $N$  if  $tf - idf(f_i) > \vartheta$ , where  $\vartheta$  is the corpus threshold
  3. Let  $R$  be the set of all ConceptNet relations which is partitioned into the relation sets  $H$  (hierarchical),  $S$  (synonymous) and  $F$  (functional).
  4. Every relation tuple  $r_{ij}(f_i, f_j) \in R$  is assigned to one of the sets  $S, F$  or  $H$  with ties being broken as  $H > S > F$
  5. Construct the ontology tree  $T(V, E)$  top-down. The root of the tree is taken as the domain name. Initially  $V = \{domain\_name\}, E = \{\emptyset\}$ .
  6. Add a vertex  $v_j$  to  $V$  and an edge  $e_{ij}(v_i, v_j)$  to  $E, \forall r_{ij}(v_i, v_j) \in H$  s.t.  $v_i \in V$  and  $v_j \in N$
  7. Merge  $v_j$  with  $v_i \forall r_{ij}(v_i, v_j) \in S$  s.t.  $v_i \in V$  and  $v_j \in N$
  8. Add a vertex  $v_j$  to  $V$  and an edge  $e_{ij}(v_i, v_j)$  to  $E, \forall r_{ij}(v_i, v_j) \in F$  s.t.  $v_i \in V$  and  $v_j \in N$
- Output:  $T(V, E)$

#### Algorithm 1. Ontology Tree Construction from ConceptNet

The *hierarchical relation set* is taken first, and the tree is constructed recursively, such that the parent concept in any hierarchical relation is already in the tree and the child concept belongs to the set of frequently occurring concepts in the domain. The *synonymous relation set* is taken next, and similar concepts are merged recursively, such that one of the concepts in any synonymous relation is already in the tree and the other concept belongs to the frequently occurring *feature set*. In the last step, the *functional relation set* is taken and processed in the same way as the hierarchical relation set.

The constructed ontology tree depicts the product attributes in the domain and the different parent-child relations. The ontology creation does not require any labeled training data. Algorithm 1 shows the detailed steps for the ontology creation. Figure 1 shows a snapshot of the constructed ontology.

### 3.2 Feature Specific Sentiment Extraction

A review or a given sentence may contain multiple features with a different opinion regarding each feature. Given a sentence and a target fea-

ture, it is essential to obtain the polarity of the sentence with respect to the feature. For example the sentence, “*The movie had a nice plot but the acting was too shabby*”, is *positive* with respect to *plot* but *negative* with respect to *acting*.

In this work, we use the feature-specific sentiment extraction approach in Mukherjee *et al.* (2012), which do not need labeled review data for training. The authors use *Dependency Parsing* to capture the association between any specific feature and the expressions of opinion that come together to describe that feature.

Given a sentence  $S$ , let  $W$  be the set of all words in the sentence. Let  $R$  be the list of *significant* dependency parsing relations (like *nsubj*, *doobj*, *advmod*, *amod* etc.), which are learnt from a corpus. A Graph  $G(W, E)$  is constructed such that any  $w_i, w_j \in W$  are directly connected by  $e_k \in E$ , if  $\exists R_i$  s.t.  $R_i(r_i, r_j) \in R$ . The *Nouns* are extracted by a POS-Tagger which form the initial feature set  $F$ . Let  $f_i \in F$  be the target feature.

We initialize ‘ $n$ ’ clusters  $C_i$ , corresponding to each feature  $f_i \in F$  s.t.  $f_i$  is the clusterhead of  $C_i$ . We assign each word  $w_i \in S$  to the cluster whose clusterhead is closest to it. The distance is measured in terms of the number of edges in the shortest path, connecting any word and a clusterhead. Any two clusters are merged if the distance between their clusterheads is less than some threshold. Finally, the set of words in the cluster  $C_i$ , corresponding to the target feature  $f_i$  gives the opinion about  $f_i$ .

The words in the cluster  $C_i$  are classified with the help of a lexicon (*majority voting*) to find the polarity  $p_i \in \{-1, 0, 1\}$  about the target feature  $f_i$ .

### 3.3 Sentiment Aggregation

Consider the camera review example in Section 1, and Figure 1 where the facets of the review are mapped to the camera ontology with their specific polarities. It can be observed that the product attributes at a higher level of the tree dominate those at the lower level. If a reviewer says something positive or negative about a particular feature, which is at a higher level in the ontology tree (say *picture*), it weighs more than the information of all its children nodes (say *light*, *resolution*, *color* and *compression*). This is because the parent feature abstracts the information of its children features. The feature importance is captured by the height of a feature node in the ontol-

ogy tree. In case the parent feature polarity is neutral, its polarity is given by its children feature polarities. Thus the information at a particular node is given by its self information and the weighted information of all its children nodes. The information propagation is done bottom-up to determine the information content of the root node, which gives the polarity of the review.

Consider the ontology tree  $T(V, E)$  where  $V_i \in V$  is a *product attribute set*. The product attribute set  $V_i$  is represented by the tuple  $V_i = \{f_i, p_i, h_i\}$ , where  $f_i$  is a product feature,  $p_i$  is the review *polarity score* with respect to  $f_i$  and  $h_i$  is the height of the *product attribute* in the *ontology tree*.  $e_{ij} \in E$  is an *attribute relation type* (Section 3.1.1) connecting  $f_i \in V_i, f_j \in V_j$  and  $V_i, V_j \in V$ . Let  $V_{ij}$  be the  $j^{\text{th}}$  child of  $V_i$ .

The *positive sentiment weight* (PSW) and *negative sentiment weight* (NSW) of a vertex  $V_i$  are defined as,

$$PSW(V_i) = h_i \times p_i^+ + \sum_j PSW(V_{ij}) \times u_{ij}$$

$$NSW(V_i) = h_i \times p_i^- + \sum_j NSW(V_{ij}) \times u_{ij}$$

where  $p_i^+ \in \{0, 1\}$  and  $p_i^- \in \{-1, 0\}$ .

The review polarity is given by the *expected sentiment-weight* (ESW) of the tree defined as,

$$ESW(\text{root}) = PSW(\text{root}) + NSW(\text{root})$$

Consider Figure 1 and assume the edge-weights of the tree to be 1.

$$PSW(\text{accessories}) = 2 \times 0 + (1 \times 1 + 1 \times 1 + 1 \times 1) = 3$$

$$NSW(\text{accessories}) = 0, PSW(\text{picture}) = 0, PSW(\text{video}) = 1$$

$$NSW(\text{picture}) = 3 \times -1 + (-1 \times 2 - 1 \times 2) = -7$$

$$PSW(\text{camera}) = 4, NSW(\text{camera}) = -7, ESW(\text{camera}) = -3$$

Figure 2 shows a snapshot of the camera ontology tree annotated with positive and negative sentiment weights. Each feature node  $f_i$  is annotated with a tuple  $[p_i^+, p_i^-]$  corresponding to its positive sentiment weight and negative sentiment weight respectively. Absence of a weight indicates that the feature node has a neutral sentiment. The figure depicts the importance of hierarchical learning as the negative sentiment weight of *picture*, at a higher level of the tree, dominates the positive sentiment weight of the other feature nodes at a lower level in the tree, resulting in the overall review polarity being negative.

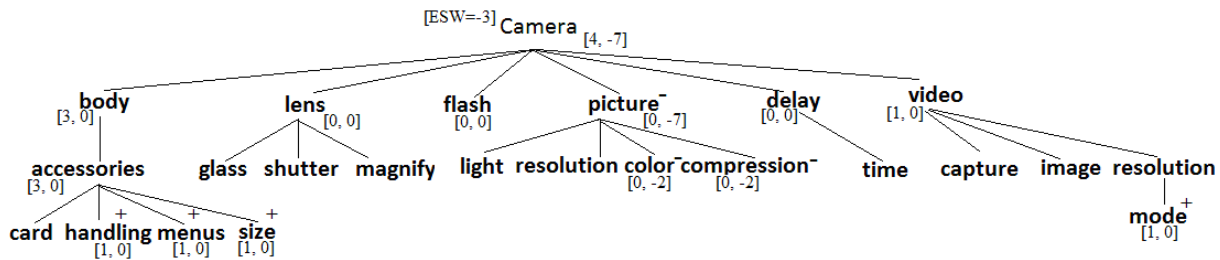


Figure 2. Snapshot of Camera Ontology Tree with Sentiment Weights

## 4 Experimental Evaluation

Analysis is performed in three domains corresponding to *automobile*, *camera* and *software*.

### 4.1 Dataset Preparation

Domain	Positive Reviews	Negative Reviews	Total Reviews
Automobile	584	152	736
Camera	986	210	1196
Software	1000	915	1915

Table 3. Dataset Statistics

The camera reviews are collected from Amazon.com and manually tagged as positive or negative. The automobile and software reviews<sup>2</sup> are taken from Blitzer *et al.* (2007). Table 3 shows the dataset statistics.

All the words are lemmatized in the reviews so that camera and cameras are reduced to the same root word camera.

Words like *hvnt*, *dnt*, *cnt*, *shant* etc. are replaced with their proper form in both our model and the baseline to capture negation.

### 4.2 Baselines

In this work, we consider three unsupervised baselines to compare the proposed approach.

**1. Lexical Baseline:** Lexical classification (Taboada *et al.*, 2011) is taken as the *first* baseline for our work. A sentiment lexicon is taken which contains a list of positive and negative terms. If the number of positive terms is greater than the number of negative terms, the review is considered to be positive and negative otherwise. The same approach is also used in our work while finding the polarity of the cluster representing the feature-specific opinion about a review. The lexical baseline considers all unigrams to be equally important, whereas we distinguish features by their position in the ontology hierarchy. This baseline model does not incorporate feature-specificity.

We experimented with three publicly available lexicons to obtain unigram polarities:

1. SentiWordNet 3.0 (Baccianella *et al.*, 2010)
2. General Inquirer (Stone *et al.*, 1966)
3. Bing Liu Lexicon (Hu *et al.*, 2004)

**2. Corpus Feature-Specific Baseline:** Tf-Idf measure is used to obtain the frequently occurring concepts in the domain from an unlabeled corpus. A feature-specific sentiment extraction model (Mukherjee *et al.*, 2012) is used to find the review polarity regarding each feature. A linear aggregation of the feature-specific polarities is done to obtain the overall review polarity. If the aggregation of the positive feature-specific polarities is greater than the aggregation of the negative feature-specific polarities, the review is considered to be positive and negative otherwise.

This model resembles the approach of LARA (Wang *et al.*, 2010) in a loose way, where the authors jointly learn the feature weights and feature-specific polarities.

**3. ConceptNet and Corpus Feature-Specific Baseline:** In this baseline, the features are extracted using ConceptNet and an unlabeled corpus using Algorithm 1. The feature set  $\bar{F} = H \vee S \vee F$  is considered and the same feature-specific sentiment extraction model is used to aggregate all the feature-specific polarities in the set.

All the baselines lack sentiment aggregation (refer Section 3.3) using ontological information.

A simple *negation* handling approach is used both in our work and the baselines. A window of size 5 (Hu *et al.*, 2004) is taken and polarities of all the words appearing in the window starting from any of the negation operators *not*, *neither*, *nor* and *no* are reversed.

Table 4 shows the three baselines and the proposed approach with the different features used in the models.

<sup>2</sup> <http://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

Models	Lexical	Corpus	ConceptNet	Sent. Aggr.
Lexical Baseline	Y			
Corpus Feature Specific Baseline	Y	Y		
Corpus and ConceptNet Feature Specific Baseline	Y	Y	Y	
Sent. Aggr. With Ontology Info.	Y	Y	Y	Y

**Table 4.** Models and Baselines

### 4.3 Results

Stanford Pos-Tagger<sup>3</sup> is used to part-of-speech tag the reviews to find the frequently occurring concepts (*Nouns*) in the domain. The ontology construction is done using ConceptNet 5<sup>4</sup>. The depth of the ontology tree is taken till level 4. The ontology depth has been empirically fixed. Further increase in depth leads to topic drift and domain concept dilution. Table 5 shows the number of frequently occurring concepts in the corpus, and the total number of nodes, leaf nodes and edges in the ontology tree for each domain.

Domains	Corpus Frequent Features	Ontology Nodes	Ontology Edges	Leaf Nodes
Automobile	268	203	202	76
Camera	768	334	333	148
Software	1020	764	763	208

**Table 5.** Ontology Tree Statistics

Table 6 shows the accuracy of the three lexical baselines in different domains in the dataset.

Lexicons	Auto-mobile	Camera	Software
SentiWordNet 3.0	60.88	59.32	60.76
General Inquirer	65.70	68.15	66.14
Bing Liu Lexicon	64.43	63.65	69.38

**Table 6.** Lexical Baselines

Stanford Dependency Parser<sup>5</sup> is used to parse the reviews for dependency extraction during feature-specific sentiment analysis (refer *Section*

<sup>3</sup> <http://nlp.stanford.edu/software/tagger.shtml>

<sup>4</sup> <http://conceptnet5.media.mit.edu/>

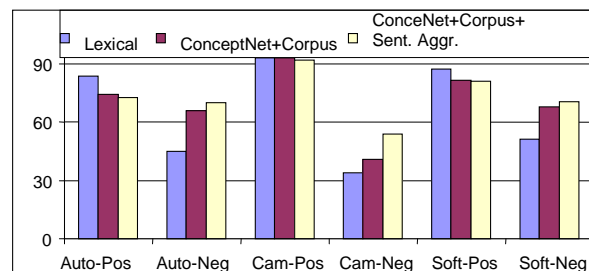
<sup>5</sup> <http://nlp.stanford.edu/software/lex-parser.shtml>

3.2). All the edge weights  $u_{ij}$  are taken to be 1. Table 7 shows the overall accuracy comparison of the proposed approach with the baselines. Bing Liu sentiment lexicon is used in all the approaches as it is found to deliver a better performance compared to the other lexicons in our model.

Models	Automobile	Camera	Software
Lexical Baseline (Bing Liu)	64.43	63.65	69.38
Corpus	68.34	65.25	72.54
ConceptNet + Corpus	70.19	67.15	74.74
ConceptNet + Corpus + Sent. Aggr.	<b>71.38</b>	<b>72.90</b>	<b>76.06</b>

**Table 7.** Overall Accuracy of All Models

Figure 3 shows the accuracy of different models on the positive and negative dataset in each domain.



**Figure 3.** Positive and Negative Accuracy of Models in Each Domain

## 5 Discussions

In this section, we discuss the observations from the experimental results of using sentiment aggregation approach with ConceptNet Ontology.

**1. Ontology Construction:** The first part of our work outlines an approach to leverage ConceptNet to construct a domain-specific ontology for product reviews. It is a difficult task to evaluate the purity of any ontology. In our work, we only perform a qualitative analysis where the constructed ontology is found to contain most of the relevant concepts in the given domain with appropriate hierarchy.

It is observed that 75.75% of the concepts in the automobile domain are mapped to some relevant concept in the corresponding product ontology; the corresponding figures for the camera and software domain being 43.49% and

74.90% respectively. In the camera domain, although the number of ontology feature nodes is much less than the frequently occurring concepts in the reviews, the proposed model performs much better than the baseline, which considers all features to be equally relevant. This shows that the ontology feature nodes capture concepts which are most relevant to the product and hence, makes a difference to the overall review polarity.

**2. Lexical Baseline Performance:** General Inquirer and Bing Liu sentiment lexicons outperform SentiWordNet in our dataset. Bing Liu sentiment lexicon was subsequently found to work better in our model than General Inquirer.

**3. Corpus Feature-Specific Baseline:** A significant accuracy improvement is observed over the lexical baseline due to the consideration of feature-specific polarities of relevant features mined from the frequently occurring concepts in the domain corpus.

**3. ConceptNet and Corpus Feature-Specific Baseline:** Incorporating ConceptNet information during the feature extraction process from the corpus improves the model performance. Only the features that frequently occur in the domain and form an important concept in the ontology hierarchy are retained.

**4. Sentiment Aggregation:** The model using sentiment aggregation approach by combining the feature-specific polarities with ontology information achieved the best accuracy in all the three domains.

**5. Negative Opinion Detection:** Reviews have much more explicit positive expressions of opinion than negative ones (Kennedy *et al.*, 2006; Voll *et al.*, 2007; Mukherjee *et al.*, 2012). This is because negative emotions are often very implicit and difficult to capture, as in sarcasm and thwarting. This is evident from Figure 3, where the lexical baseline attains a high accuracy on positive reviews in all the domains, but fares very poorly on negative reviews. The other two models, on the other hand, perform much better on the negative reviews. This shows that the ontology based sentiment extraction method is able to capture negative sentiment much more strongly. The model also paves the way for analyzing reviews which contain more positive expressions of opinion than negative ones, but are still tagged as negative;

which cannot be captured by a feature-counting classifier.

**6. Sentiment Ontology Tree Personalization:** In this work, we have assumed all relations to be equally important, and thus considered the edge weights in the tree to be 1. However, the model allows the ontology tree to be personalized to suit the purpose of an individual and incorporate subjectivity in the reviews. If an individual prefers functional relations or use of certain features over its components, this information can be incorporated in the tree. This allows the general domain-specific ontology tree to be customized to an individual's interest.

## 6 Conclusions and Future Work

In this work, we outline an approach to combine the feature-specific polarities of a review with ontology information to give better sentiment classification accuracy. The proposed approach leverages ConceptNet to automatically construct a domain specific ontology tree. We performed experiments in multiple domains to show the performance improvement induced by the sentiment aggregation approach using ontology information over simple aggregation of feature-specific polarities.

The work is mostly unsupervised, requiring no labeled training reviews. The performance of the classifier is subject to the coverage of the lexicon and the accuracy of the feature-specific classifier.

The work also addresses the idea of personalizing a sentiment ontology tree to suit an individual's interest over specific features and parent-feature relations. This is also the first work, to the best of our knowledge, to discuss an approach to deal with reviews having majority positive (or negative) features but still tagged as negative (or positive). Reviews, of such kind, can be aptly handled using ontology information which captures the intrinsic specificities of product-feature relations in a given product domain.

## References

- A. Sureka, V. Goyal, D. Correa, and A. Mondal. 2010. Generating Domain-Specific Ontology from Common-Sense Semantic Network for Target-Specific Sentiment Analysis. In Fifth International Conference of the Global WordNet Association (GWC).
- Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie and product reviews using

- contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006.
- Chen Mosha. 2010. Combining Dependency Parsing with Shallow Semantic Analysis for Chinese Opinion-Element Relation Identification. *IEEE*, pp.299-305.
- Dustin Smith and Stan Thomas. 2004. Investigating ConceptNet. Ph.D Thesis. December 2004
- Estival, D. Nowak, C. and Zschorn, A. 2004. Towards Ontology-Based Natural Language Processing. In *Proceedings of 4th Workshop on NLP and XML*.
- Hu, Mingqing and Liu, Bing. 2004. Mining and summarizing customer reviews, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Seattle, Washington, USA.
- Hugo Liu and Push Singh. 2004. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal*, 22(4):211–226, October.
- Jaap Kamps, Maarten Marx, Robert J. Mokken, and Marten de Rijke. 2002. Words with attitude. In *Proceedings of the 1st International Conference on Global WordNet*, Mysore, India
- John Blitzer, Mark Dredze, Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of Association of Computational Linguistics (ACL)*.
- Kimberly Voll and Maite Taboada. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence*.
- Kotov, Alexander and Zhai, ChengXiang. 2012. Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries. In *Proceedings of the fifth ACM international conference on Web search and data mining*.
- Lin, Chenghua and He, Yulan. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM)*.
- Mingqing 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*.
- P.D. Turney and M.L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.
- P.D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia
- Sergio Guadarrama and Marta Garrido. 2008. Concept-Analyzer: A tool for analyzing fuzzy concepts. In *Proceedings of IPMU*.
- Stefano Baccianella and Andrea Esuli and Fabrizio Sebastiani. 2010. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, *LREC*.
- Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M. and associates. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.
- Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. Part 1, *Lecture Notes in Computer Science*, Springer 7181:475–487.
- Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. WikiSent: Weakly Supervised Sentiment Analysis through Extractive Summarization with Wikipedia. In *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference*.
- Taboada, Maite and Brooke, Julian and Tofiloski, Milan and Voll, Kimberly and Stede, Manfred. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*.
- Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP 2004*, pp.412-418.
- V. Hatzivassiloglou and J. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity.
- V. Hatzivassiloglou and K.R. McKeown. 2002. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the ACL*.
- Yuanbin Wu, Qi Zhang, Xuanjing Huang, Lide Wu. 2009. Phrase Dependency Parsing for Opinion Mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Wang, Hongning and Lu, Yue and Zhai, ChengXiang. 2010. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*
- Wei, Wei and Gulla, Jon Atle. 2010. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.



# Detecting Cyberbullying Entries on Informal School Websites Based on Category Relevance Maximization

Taisei Nitta † Fumito Masui † Michal Ptaszynski † Yasutomo Kimura §  
Rafal Rzepka ‡ Kenji Araki ‡

† Department of Computer Science, Kitami Institute of Technology

nitta@ialab.cs.kitami-it.ac.jp, {f-masui,ptaszynski}@cs.kitami-it.ac.jp

§ Department of Information and Management Science, Otaru University of Commerce

kimura@res.otaru-uc.ac.jp

‡ Graduate School of Information Science and Technology, Hokkaido University

{kabura,araki}@media.eng.hokudai.ac.jp

## Abstract

We propose a novel method to detect cyberbullying entries on the Internet. “Cyberbullying” is defined as humiliating and slandering behavior towards other people through Internet services, such as BBS, Twitter or e-mails. In Japan members of Parent-Teacher Association (PTA) perform manual Web site monitoring called “net-patrol” to stop such activities. Unfortunately, reading through the whole Web manually is an uphill task. We propose a method of automatic detection of cyberbullying entries. In the proposed method we first use seed words from three categories to calculate semantic orientation score PMI-IR and then maximize the relevance of categories. In the experiment we checked the cases where the test data contains 50% (laboratory condition) and 12% (real world condition) of cyberbullying entries. In both cases the proposed method outperformed baseline settings.

## 1 Introduction

“Cyberbullying” is a new form of bullying. It is carried out on the Internet instead of classrooms. It takes form of hate messages sent through e-mails, electronic Bulletin Board System (BBS), etc., with the use of personal computers or mobile phones. Recently it has become a serious social problem in many countries, one of which is Japan (MEXT, 2008). Examples of cyberbullying that actually happened, include ridiculing student personality, body type, or appearance on informal school BBS, slandering students and insinuating they had performed deviate sexual intercourses. Some cases of cyberbullying lead the students who were bullied to assault or kill themselves or the student who wrote the bullying entry on the BBS.

To deal with the problem members of Parent-Teacher Association (PTA)<sup>1</sup> perform Web site monitoring activities called “net-patrol”. When a harmful entry is detected the net-patrol member who found it sends a request to remove the entry to the Internet provider or Web site administrator. Some of the actual examples of harmful entries which were requested for deletion are represented in Table 1 (names, phone numbers and other personal information was changed).

Unfortunately, net-patrol has been carried out mostly manually. It takes much time and effort to find harmful entries (entries that contain harmful information and expressions) in a large amount of contents appearing on countless number of bulletin board pages. Moreover, the task comes with a great psychological burden on mental health to the net-patrol members. To solve the above problem and decrease the burden of net-patrol members, Matsuba et al. (2011) proposed a method to detect harmful entries automatically.

In their method they extended the method of relevance calculation PMI-IR, developed by Turney (2002) to calculate relevance of a document with harmful contents. With the use of a small number of seed words they were able to detect effectively large numbers of document candidates for harmful entries.

Their method was proved to determine harmful entries with an accuracy of 83% on test data for which about a half contained harmful entries. However, it was not yet verified how well the method would perform in real life conditions, where the ratio of cyberbullying entries and normal contents is not equal.

In this research, based on Matsuba et al.’s method of obtaining maximal relevance values for seed words, we propose a method for maximization of relevance score of seed words. In our

<sup>1</sup>An organization composed of parents, teachers and school personnel.



method we divide seed words into multiple categories and calculate maximal relevance value for each seed word with each category. By calculating the score, representing semantic orientation of “harmfulness”, the method is expected to detect harmful entries more effectively than in the previous research. Moreover, we evaluate our method on data sets with different ratios of harmful contents to verify the usability of the method in the most realistic way.

The paper outline is as follows. Firstly, we describe research on extraction of harmful entries in Section 2. Next, we describe the proposed method in Section 3. Furthermore, in Section 4 we construct evaluation data sets with different ratios of cyberbullying entries, perform evaluation experiments based on these data sets, and describe the results of the experiments. We present a discussion and explain the results in detail in Section 5. Finally, in Section 6 we conclude the paper and propose some of the ideas for future improvement of the method.

## 2 Related research

There has been a number of research on extracting harmful information before. For example, Ishisaka and Yamamoto (2010) have focused on developing an abusive expression dictionary based on a large Japanese electronic bulletin board “2 channel”. In their research Ishisaka and Yamamoto firstly defined words and paragraphs in which the speaker directly insults or slanders other people with the use of explicit words and phrases such as バカ (*baka*) “stupid”, or マスゴミのクズ (*masugomi no kuzu*) “trash of mass-mudia”. Next, they studied the use of abusive language, in particular which words appear the most often with abusive expressions, and based on this study they extracted abusive expressions from the surrounding context.

In other research Ikeda and Yanagihara (2010) have manually collected and divided separate sentences into harmful and non-harmful, and based on word occurrence within the corpus they created a list of keywords for classification of harmful contents. Next they utilized context of dependency structures of sentences containing harmful and non-harmful contents to improve the system performance. However, on the Web there are numerous variations of the same expressions dif-

<sup>2</sup>*Mobage* and *Gree* are online game service Web sites.

Table 1: Examples of harmful entries which were requested for deletion. Japanese (above), transliteration (middle), English translation (below).

- 調子乗りすぎいつべん殺らなあかんで ( <i>Chōshi nori sugi ippen yara na akan de</i> ) “Don’t get excited that much or I’ll kill ya!”
- 新田キモイつかキショイほんま死んで ( <i>Nitta kimoi tsuka kishoi honma shinde</i> ) “Nitta [proper noun], you’re ugly, or rather fugly, just die, man”
- ンな奴どつき回したれ ( <i>N’na yatsu dotsuki mawashi tare</i> ) “What an ass, slap him”
- 性格わるーい ぶちやいく一笑 ( <i>Seikaku waru-i buchaiku-warai</i> ) “Baaad personality, and an ugly hag, lol”
- >> 17 あの女、昔、モバだったかグりに登録してたヤリマンじゃん。 ( <i>Ano onna mukashi Moba dattaka Guri ni tōroku shiteta yariman jan</i> ) “>> 17 that woman is the same one who was bitching around before on <i>Mobage</i> or <i>Gree</i> .”
- すぐにヤレる。01234567890。 めっちゃカワイイで ( <i>Sugu ni yareru. 01234567890. Meccha kawaii de.</i> ) “You can take her out even now. 01234567890. She’s a great lay.”

fering with only one or two characters, such as 爆破 (*bakuha*) “blow up” and 爆一破 (*baku-ha*) “blooow up”. The weakness of this method is that all of the variations of the same expression need to be collected manually, which is very time-consuming.

Fujii et al. (2010) proposed a system for detecting documents containing excessive sexual descriptions using a distance between two words in a sentence. In their method they determine as harmful those words which are in closer distance to words appearing only in harmful context (“black words”) rather than those in closer distance to words which appear in both harmful and non-harmful context (“grey words”).

Hashimoto et al. (2010) proposed a method for detecting harmful meaning in jargon. In their method they assumed that the non-standard meaning is determined by the words surrounding the

word in question. They detected the harmful meaning based on calculating co-occurrence of a word with its surrounding words.

In our research we did not consider the surrounding words. Instead, we determine the harmfulness of input by calculating the harmfulness score for all word sequences in input. Moreover, since we check the co-occurrence of word sequences on the Web, our method greatly reduces the cost of manual construction of training data. Furthermore, in calculating the harmfulness score we apply dependency relations between phrases. Therefore there is no need to check all words proceeding and succeeding the queried words, which greatly reduces processing time.

### 3 Proposed method

In this section we present an overview of the method for maximization of category relevance. In the proposed method we extend the method proposed by Turney (2002) to calculate the relevance of seed words with entries from the bulletin board pages. Moreover, we apply multiple categories of harmful words and calculate the degree of association separately for each category. Finally, as the harmfulness score (or polarity of “harmfulness”) we choose the maximum value achieved by all categories. The method consists of three steps. (1) Phrase extraction, (2) Categorization and harmful word detection together with harmfulness polarity determination, (3) Relevance maximization. Each of the steps is explained in detail in the following paragraphs.

#### 3.1 Phrase extraction

In cyberbullying entries the harmful character of an entry can be determined by looking at separate words. For other cases however, even if a word in itself is not harmful, it gains harmful meaning when used in a specific context, or in combination with other words. For example, for a pair of words 性格が悪い (*seikaku ga warui*) “bad personality”, neither “bad”, nor “personality” on their own express harmful meaning. However, when these words are used together in a dependency relation, they become harmful (negative depiction of a person’s personality). Therefore, methods for detecting harmful contents using separate words only, will fail when they encounter an entry which gained harmful meaning by phrases containing words in dependency relation.

Table 2: Types of phrases applied in the proposed method with examples.

Phrase	Example
noun-noun	サル顔 ( <i>sarugao</i> ) “monkey face” →Description ridiculing person’s features
noun-verb	新田を殺す ( <i>Nitta wo korosu</i> ) “Kill Nitta” →Threatening expressions
noun-adjective	性格が悪い ( <i>seikaku ga warui</i> ) “bad personality” →Description criticizing person’s features

To solve this issue, we use the polarity calculation score for the morphemes<sup>3</sup> combined in the dependency relation. We define such a combination as a “phrase”. One phrase consists of a morpheme pair in dependency relation. The dependency relation is calculated using a standard morphological analyzer for Japanese (MeCab<sup>4</sup>) and a Dependency parser for Japanese (Cabocha<sup>5</sup>). The phrases defined this way are extracted from all target entries.

#### 3.2 Harmful word detection and categorization

In this process we detect words of potential harmful connotations, or “harmful words”. Harmful words often include newly coined words or informal modifications of normal transcriptions, thus are not recognized by standard preprocessing tools, such as morphological analysers or dependency parsers. Therefore, it is possible such words, unless specifically annotated, would not be handled properly and cause error in morphological analysis. We investigated the entries of informal school Websites using the definition of harmful words proposed by the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT, 2008, later referred to as “Min-

<sup>3</sup>In this report, we use the words “word” and “morpheme” in the same meaning.

<sup>4</sup><http://mecab.sourceforge.net/>

<sup>5</sup><http://code.google.com/p/cabocha/>

istry of Education”), and registered 255 harmful words (nouns, adjectives and verbs) in the dictionary of morphological analyzer. In addition, we categorized harmful words into three categories: obscene, violent and abusive. The Ministry of Education defined words considered as cyberbullying to include obscene, violent, or abusive words used on BBS. In this study, we applied the above definition, and therefore we also classified harmful words into the three categories: obscene, violence, or abusive. Next, we selected from each category three most often occurring words as seed words and registered them in the system. The words we selected include セックス (*sekkusu*) “[to have] sex”, ヤリマン (*yariman*) “slut”, フェラ (*fera*) “fellatio” for the obscene category, 死ぬ (*shinu*) “die [imperative]”, 殺す (*korosu*) “[to] kill”, 殴る (*naguru*) “[to] slap” for the violent category, and うざい (*uzai*) “annoying”, きもい (*kimoi*) “gross”, 不細工 (*busaiku*) “ugly” for the abusive category.

### 3.3 Maximization of relevance score

In this process we calculate harmfulness polarity score of phrases with each seed word for all three categories. We use pointwise mutual information (PMI) score as a measure of relevance between a phrase and harmfulness polarity words from each category. PMI here indicates a co-occurrence frequency of the queried phrase with the three words registered for each category. To calculate the co-occurrence frequency we use information retrieval (IR) score. Countless number of various pages exists on the Web, and thus various words are written there. Therefore, it is possible to obtain a high coverage by using the IR score.

We calculate the relevance of a phrase with words from each category according to the following equation (1).  $p_i$  is a phrase extracted from the entry,  $w_j$  are three words that are registered in one category of harmfulness polarity words,  $hits(p_i)$  and  $hits(w_j)$  are Web search hits for each category for  $p_i$  and  $w_j$  respectively,  $hits(p_i \& w_j)$  is a number of hits when  $p_i$  and  $w_j$  appear on the same web page. Finally,  $PMI - IR(p_i, w_j)$  is the relevance of  $p_i$  and  $w_j$ .

$$PMI - IR(p_i, w_j) = \log_2 \left\{ \frac{hits(p_i \& w_j)}{hits(p_i)hits(w_j)} \right\} \quad (1)$$

From all three scores calculated for the phrase with seed words from the three categories, we select the category which achieved the highest score

as the one of the highest relevance with the phrase. We calculate the relevance score this way for all phrases extracted from the entry. Finally we select the category with the maximal overall score as the one with the highest relevance with the entry. The *score* is calculated according to the following equation (2).

$$score = \max(\max(PMI - IR(p_i, w_j))) \quad (2)$$

In the baseline settings (Matsuba et al., 2011) the relevance was calculated as a sum of all scores for all phrases with each harmful word separately. In this method instead of taking all words separately we group them in categories and calculate the relevance with three most common harmful words from each category. By incorporating this improvement during the Web search the retrieved pages are those for which the phrase appeared not only with one of the harmful words, but with all three words from one category. This not only reduces the processing time, but also improves the calculation of the relevance score, since only the strongest (most harmful) phrases are selected. Moreover, since it is easier to find a Web page containing a phrase and the only one harmful word, than the phrase with three words together, calculating the relevance for all harmful words separately allowed phrases with low actual relevance to achieve high scores in the baseline system. Maximization of the relevance *score* prevents low relevance phrases to erroneously achieve high scores.

We explain this process on the following example: 可愛いけど性格が悪い女 (*kawaii kedo seikaku ga warui onna*) “Cute girl, but bad personality”. Firstly, from the entry the extracted phrases are: 可愛い-女 (*kawaii-onna*) “cute-girl”, 性格-悪い (*seikaku-warui*) “bad-personality”, 悪い-女 (*warui-onna*) “bad-girl”. Next, we calculate the relevance between “cute-girl” and the three groups of words separately (“sex, slut, fellatio”, “die, kill, slap”, “annoying, gross, ugly”). The highest maximal score is selected as the relevance (harmfulness score) of this phrase. Similarly the score is calculated for “bad-personality” and “bad-girl”. From all the scores for all phrases the highest overall score is considered as the maximized harmfulness *score* of the phrase. All entries are then sorted beginning with the one with the highest harmfulness score. Finally, we set a

Table 3: The numbers of entries on informal school BBS including the number and percentage of cyberbullying entries.

BBS	Overall number of entries	Cyberbullying entries	Percentage (%)
BBS(1)	600	75	12.5
BBS(2)	736	90	12.2
BBS(3)	886	100	11.3

harmfulness threshold  $n$  and consider  $n$  entries with the highest score as harmful, and discard other as irrelevant to check how many of the entries within the specified threshold are in fact harmful.

## 4 Evaluation experiment

We performed an experiment to evaluate the performance of the proposed method, and compared the results with the baseline. Below we describe the preliminary study for carrying out the experiment (Section 4.1), explain the experiment settings (Section 4.2) and report the results of experiments (Section 4.3).

### 4.1 Preliminary study

It is necessary to create a test data with harmful and non-harmful entries mixed at an appropriate rate. In previous research the mixing was set at the same rate (half of the entries included cyberbullying). However, it cannot be assumed that harmful entries appear in real life with the same rate as normal ones. Therefore to evaluate our method in conditions closer to reality we performed a preliminary study to verify how much of the entries are harmful on actual Web pages. We counted the harmful entries mixing ratio on three informal school Websites, in particular we focused on informal school bulletin boards (BBS). The result of the study is represented in Table 3. We performed the study during four days between January 27 and 30, 2012. The number of obtained entries was 2,222.

As of the result of the study, the first BBS contained a total number of 600 entries from which 75 were harmful, which indicates that harmful entry appearance rate was 12.5%. Similarly, for the second BBS, 90 out of 736 total entries were harmful (12.2%). On the third BBS there were 886 total entries with 100 harmful ones (11.3%). From the above results, we concluded that about 12%

of all entries appearing on informal school BBS can be accounted as cyberbullying. Therefore in the experiment we verified the performance of the method under the condition when harmful entries cover 12% of the whole data.

### 4.2 Experiment settings

In the evaluation experiment we compared the performance of the proposed method to the baseline. We did this firstly for the case where the test data contained 50% of harmful entries. Next, we prepared a different test data, which contained 12% of harmful entries and compared the performance under this condition for both the baseline and the proposed method.

The test data containing 50% of harmful entries contains 2,998 entries in total with 1,508 of harmful entries and 1,490 of non-harmful entries. The dataset contains actual collection gathered by the net-patrol members from bulletin boards, and additional data gathered manually by Matsuba et al. (2011) (the latter were collected from the BBS sites limited to schools from the Mie Prefecture, Japan). We performed a 10-fold cross validation on this dataset. We processed the dataset by both the baseline and the proposed method and calculated the harmful polarity score for entries where the phrases could be extracted. Then we ranked all entries decreasingly according to the harmful polarity score, and evaluated the performance looking at the top  $n$  entries by increasing the threshold of  $n$  by 50 each time.

To prepare the dataset for the real world condition (12% of harmful entries), we prepared five test sets by randomly extracting 60 harmful and 440 non-harmful, 500 entries in total from the original dataset. On these datasets we did not perform a 10-fold cross validation, since it would make the results not statistically relevant (each set for cross validation would contain only 60 harmful entries). Instead we calculated the results for each of the five sets separately. This allowed us to include all entries from the original test set in the evaluation. We processed these datasets with both the baseline and the proposed method and calculated the harmfulness polarity score for entries for which the phrases could be extracted. Then, similarly to the original dataset, we ranked all entries based on the harmfulness polarity score, and evaluated the performance by taking the top  $n$  and increasing the threshold  $n$  by 10 each time.

We considered automatic setting of the threshold using machine learning methods, however it was difficult due to the small size of test data for the real world condition. In the future for automatic threshold setting we plan to develop a machine learning method capable of handling small sized data. Therefore this time we increased the threshold manually each time and investigated Precision and Recall for each threshold.

As evaluation criteria we used Precision (P) and Recall (R), calculated according to the equations (3) and (4). The Precision is a ratio of the number of entries that could be properly determined as harmful to the number of all entries determined as harmful among the top  $n$ . The Recall is a ratio of the number of entries that could be properly determined to be harmful to the overall number of harmful entries. The final performance is calculated as an average of Recall and Precision for each test data in this experiment.

$$P = \frac{\text{correct annotations}}{\text{all system annotations}} \quad (3)$$

$$R = \frac{\text{correct annotations}}{\text{all harmful annotations}} \quad (4)$$

### 4.3 Results

The results showing Precision and Recall for both the baseline and the proposed method for both datasets (50% and 12% of cyberbullying entries) are represented in Figure 1. The horizontal axis and the vertical axis represent percentage of Recall and Precision for each threshold, respectively.

For the test data containing 50% of harmful entries, Precision was between 49% - 79% for the baseline, while for the proposed method Precision was between 49% - 88%.

For the test data containing 12% of harmful entries, Precision was between 11% - 31% for the baseline, while for the proposed method Precision was between 10% - 61%.

## 5 Discussion

The experiment results showed that the proposed method achieved higher overall performance comparing to the baseline. The shape of the correlation curve for Recall and Precision shows that that performance for the baseline is significantly reduced

in general comparing the test data containing 12% of harmful entries to the test data containing 50% of harmful entries. However, for the proposed method, although the performance is reduced as well, there is no sudden drop in the shape of the correlation curve when comparing both datasets.

This could suggest that the performance is more stable in the proposed method than in the baseline. There were several cases of threshold  $n$  where the Recall was slightly higher for the baseline than for the proposed method in the test data containing 12% of harmful entries. This happened because for some harmful entries the harmfulness score could not be calculated highly enough due to the fact that the score calculation is more strict (Precision-oriented) in the proposed method. Therefore, although Recall is slightly higher in the baseline for large thresholds, the Recall is higher for the proposed method for small and medium thresholds, with the Precision being constantly higher for the proposed method. Therefore, it can be said that the proposed method achieved higher general performance than the baseline.

Next we explain the results for the test data containing 12% of harmful entries. We investigated the threshold cases of entries where the Precision reaches 48%. Entries found there included, for example “アトピーのやつ死ぬよ” (*atopii no yatsu shine yo*) “The bastard with atopy must die” and “ウザイキモイぶす” (*uzai kimoi busu*) “annoying, gross and ugly”. From those entries high relevance score was calculated for phrases like “アトピー-死ぬ” (*atopii-shine*) “atopy-die” and “ウザイ-ぶす” (*uzai-busu*) “annoying-ugly”. Since the phrases included seed words as well, this most probably increased the polarity value of the harmful entry.

On the other hand, there were many non-harmful entries classified as harmful with harmfulness polarity score equally high or higher than the actual harmful entries. An example of a non-harmful entry of this kind was “県外に住んでいる” (*kengai ni sun de iru*) “living outside of the prefecture”. The phrase extracted from this entry was “外-住ん” (*soto-sun*) “outside-live”. This is a neutral phrase, and appears similarly often in non-harmful entries as well as in harmful entries in cases of exposing personal information about where a person lives. Therefore, the relevance of a harmful entry containing such a phrase is increased, and as a result overall harmfulness po-

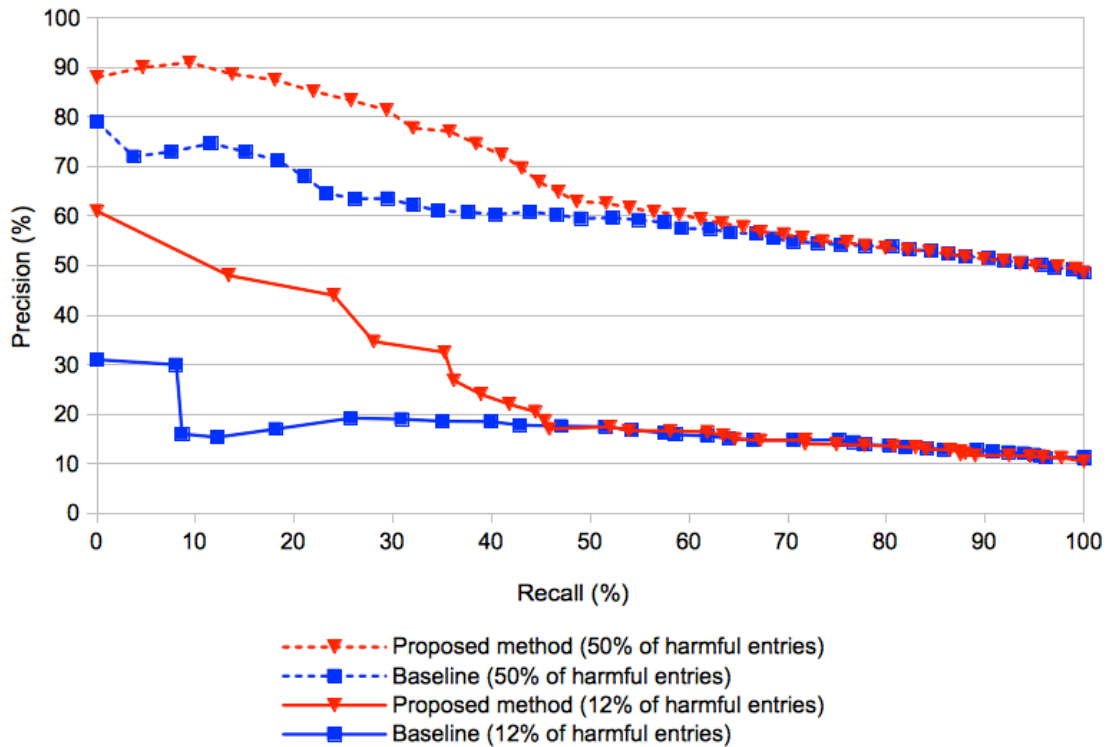


Figure 1: Precision and Recall for both the baseline and the proposed method for both datasets (50% and 12% of cyberbullying entries). The horizontal axis and the vertical axis represent percentage of Recall and Precision for each threshold, respectively. All results statistically significant. For the 50% data the results were extremely statistically significant on 0.0001 level. For the 12% data the results were statistically significant on 0.05 level.

larity score of such non-harmful entries becomes higher.

As a countermeasure it could be considered to register some words, which appear only in non-harmful entries, like “splendid”, with non-harmful polarity and calculate the relevance of non-harmful entries as well. In particular, firstly, the non-harmful words could be registered in the dictionary. Then the relevance could be calculated for the phrases with both, the non-harmful polarity words and harmful polarity words. In such cases the phrase could be considered as non-harmful when the relevance score of the phrase with non-harmful words was higher than the relevance with the harmful words. This could reduce the influence of neutral phrases on the overall performance.

We also investigated the cases where Recall reaches 100%. These cases include entries which contain personal information only such as school names or person’s names, such as “Nitta of Kitami Institute of Technology, 4th grade”, etc. The relevance of such entries with harmful words reg-

istered at present in the dictionary is low, which influenced their overall harmfulness score as well. To solve this problem we plan to register in the dictionary words which have high relevance with personal information and use them in the relevance score calculation as well.

## 6 Conclusions and future work

In this study, we proposed a method of maximization of category relevance to automatically detect cyberbullying entries on the Internet. With this research we wish to contribute to reducing the burden of Internet patrol personnel who make efforts to manually detect harmful entries appearing on the Internet. In order to verify the actual usefulness of the proposed method we evaluated the performance for the test data containing similar percentage of harmful entries as in reality. Firstly, in a preliminary study we verified the usual ratio of harmful entries on the Internet. Next, we prepared test datasets containing the same amount of cyberbullying entries as in reality and evaluated the method on these test sets. In addition, we re-

produced the baseline method and compared the performance to the proposed method.

The experiment results showed that the proposed method obtained higher results than the baseline. Under the fair condition (test dataset with 50% of harmful entries) the proposed method achieved over 90% of Precision at 10% Recall and keeping up high Precision (80-70%) at Recall close to 50%. Under the real world condition (test dataset with 12% of harmful entries) the method achieved nearly 50% of Precision at about 10% of Recall. The relevance curve have decreased slowly with growing Recall for the proposed method, while for the baseline the relevance curve has dropped suddenly from 30% to around 15% at the same Recall rate. As for drawbacks in our method, harmful entries consisting of personal information were scored as less harmful due to the appearance of neutral phrases which appear often in both harmful and non-harmful entries.

In the near future, we plan to register non-harmful polarity words which have a high relevance with non-harmful entries to lower the overall harmfulness polarity score of non-harmful entries containing neutral phrases. We will also investigate a method for assessing the harmfulness score to entries including personal information. Furthermore, we plan to increase the data set, and determine the optimal threshold automatically by using machine learning.

## Acknowledgement

This work was supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research) Grant Number 24600001.

## References

Ministry of Education, Culture, Sports, Science and Technology (MEXT). 2008. *'Netto-jō no ijime' ni kansuru taiō manyuaru jirei shū (gakkō, kyōin muke)* ["Bullying on the Net" Manual for handling and collection of cases (for schools and teachers)] (in Japanese). Published by MEXT.

Tatsuaki Matsuba, Fumito Masui, Atsuo Kawai, Naoki Isu. 2001. *Gakkō hi-kōshiki saito ni okeru yūgai jōhō kenshutsu wo mokuteki to shita kyokusei hantei moderu ni kansuru kenkyū* [Study on the polarity classification model for the purpose of detecting harmful information on informal school sites] (in Japanese), In *Proceedings of The Seventeenth Annual Meeting of The Association for Natural Language Processing (NLP2011)*, pp. 388-391.

Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, pp. 417-424.

Tatsuya Ishisaka, Kazuhide Yamamoto. 2010. *2chaeru wo taishō to shita waruguchi hyōgen no chūshutsu* [Extraction of abusive expressions from 2channel] (in Japanese), In *Proceedings of The Sixteenth Annual Meeting of The Association for Natural Language Processing (NLP2010)*, pp.178-181.

Kazushi Ikeda, Tadashi Yanagihara. 2010. *Kakuyōso no chūshōka ni motozuku ihō-, yūgai-bunsho kenshutsu shuhō no teian to hyōka* [Proposal and evaluation of method for illegal and harmful document detection based on the abstraction of case elements] (in Japanese), In *Proceedings of 72nd National Convention of Information Processing Society of Japan (IPSJ72)*, pp.71-72.

Yutaro Fujii, Satoshi Ando, Takayuki Ito. 2010. *Yūgai jōhō firutaringu no tame no 2-tango-kan no kyori oyobi kyōki jōhō ni yoru bunshō bunrui shuhō no teian* [Developing a method based on 2-word co-occurrence information for filter harmful information] (in Japanese), In *Proceedings of The 24th Annual Conference of The Japanese Society for Artificial Intelligence (JSAI2010)*, paper ID: 3D2-4, pp. 1-4.

Hiroshi Hashimoto, Takanori Kinoshita, Minoru Harada. 2010. *Firutaringu no tame no ingo no yūgai goi kenshutsu kinō no imi kaiseki shisutemu SAGE e no kumikomi* [The function that detect harmful word sense from slang built into the semantic analysis system SAGE for filtering] (in Japanese), IPSJ SIG Notes 2010-SLP-81(14), pp. 1-6.

# A Lexicon-based Investigation of Research Issues in Japanese Factuality Analysis

Kazuya Narita\*, Junta Mizuno<sup>†</sup> and Kentaro Inui\*

\*Graduate School of Information Sciences, Tohoku University, Japan

<sup>†</sup>National Institute of Information and Communications Technology (NICT), Japan

{narita, junta-m, inui}@ecei.tohoku.ac.jp

## Abstract

Event factuality is information about whether events mentioned in natural language correspond to either actual events that have occurred in the real world or events that are of uncertain interpretation. Factuality analysis is useful for information extraction and textual entailment recognition, among others, but sufficient performance has not yet been achieved by the machine learning-based approach. It is now important to take a closer look at the linguistics phenomena involved in factuality analysis and identify the technical research issues more precisely. In this paper, we discuss issues regarding lexical knowledge through error analysis of a Japanese factuality analyzer based on lexical knowledge and compositionality.

## 1 Introduction

Event factuality is information about whether events mentioned in natural language correspond to either actual events that have occurred in the real world or events that are of uncertain interpretation.

- (1) a. 彼はさきほど部屋を出た。  
*kare-wa sakihodo heya-wo de-ta.*  
(He left the room a little while ago.)
- b. もう遅いから、彼は先に帰ったのだろう。  
*mou osoi-kara, kare-wa saki-ni kaet-ta-no-daro-u.*  
(It's late now, so he may have gone home.)
- c. 問題が発生するのを防いだ。  
*mondai-ga hassei-suru-no-wo fusei-da.*  
(We prevented the occurrence of the problem.)

For example, we can interpret that the event “*de*” (*leave*) in (1a) is factual in the real world, the event “*kaet*” (*go home*) in (1b) is possibly factual because of the modal auxiliary “*-ta-no-daro-u*” (*may have -ed*), and the event “*hassei-suru*”

(*occurrence*) in (1c) is counterfactual because of the implicative predicate “*fusei-da*” (*prevented*).

Factuality analysis is useful for a broad range of NLP applications such as information extraction, question answering, and textual entailment recognition. Prior work on factuality analysis has made considerable efforts for designing and creating corpora manually annotated with factuality-related information (Saurí and Pustejovsky, 2009; Matsuyoshi et al., 2010; Tanaka et al., 2013, etc.) and several empirical studies on those resources are reported revealing the difficulties of the task (Inui et al., 2008; Matsuyoshi et al., 2010; Morante and Blanco, 2012; Saurí and Pustejovsky, 2012). For Japanese, Matsuyoshi et al. (2010) report that their factuality classes are highly skewed and the minority classes are very difficult for their machine learning-based models to precisely identify. The minority classes include uncertain statements as in example (1b) and counterfactual statements as in (1c). Such “marked” statements are far less frequent than unmarked statements (i.e. certain factual statements) and thus are not as easy to collect as unmarked statements. While the label distribution is reported to be less skewed in English (Szarvas et al., 2008), still uncertain and counterfactual statements constitute minority classes. In addition, uncertain and counterfactual statements exhibit a very broad variety of linguistic devices for expressing uncertainty and negation. For those reasons, the whole task is not as easy as it appears and simple strategies based on supervised machine learning do not work well.

Given this background, rather than putting everything simply into a machine learning algorithm, it is now important to take a closer look at the linguistics phenomena involved in factuality analysis and identify the technical research issues more precisely. One promising way for it is to make use of existing lexical resources and divide the whole issues into those related to lexical knowledge and the rest. We take this approach in this



paper because (i) the factuality status is primarily expressed by lexical devices such as auxiliaries (e.g. “*-ta-no-darou*” (*may have -ed*)) and factual and counterfactual predicates (e.g. “*fusegu*” (*prevent*)), and (ii) there are existing Japanese lexicons of such factuality-related expressions (factuality markers, henceforth) available with a reasonably broad coverage. As a platform for computing factuality with factuality markers, we adopt Saurí and Pustejovsky’s rule-based model for English factuality analysis (Saurí and Pustejovsky, 2012) and adapt it to the Japanese language. Saurí and Pustejovsky’s model is suitable as it assumes the availability of a factuality lexicon and uses it to identify the factuality status of each subordinate event in a compositional manner from the factuality status of its superordinate event. For lexical resources, we use the dictionary of Japanese functional expressions (Matsuyoshi et al., 2007) and the dictionary of Japanese clue expressions for extended modality (Eguchi et al., 2010). This paper presents a *first* comprehensive investigation in Japanese factuality analysis, which is based on these sufficient lexicons.

This paper is organized as follows. Section 2 describes related work. In Section 3, we construct a Japanese factuality analyzer based on compositional approach by Saurí and Pustejovsky (2012). In Section 4, we discuss issues regarding lexical knowledge through error analysis by applying our analyzer with Japanese text. Based on the analysis in Section 4, Section 5 discusses lexicon-based scope detection. Section 6 concludes this paper.

## 2 Related work

Previous work for an annotation schema of factuality and other associated information includes FactBank (Saurí and Pustejovsky, 2009), Japanese corpus with extended modality (Matsuyoshi et al., 2010), and so on. Saurí and Pustejovsky annotate event mentions with its source, epistemic modality (certainty) and polarity for representing the event factuality. Additionally, their FactBank is extended with pragmatically informed factuality judgments by de Marneffe et al. (2012). Matsuyoshi et al. mark up an event mention with seven components (*source, time, conditional, primary modality type, actuality, evaluation, and focus*). Our factuality corresponds to *actuality*. Tanaka et al. (2013) annotate the sense and usage of ambiguous expressions related to factuality.

For automatically analyzing factuality in text, there are approaches based on machine learning.

Inui et al. (2008) have proposed a method of analyzing modality and polarity of event mentions in Japanese text with an approach based on conditional random field. However, it is very difficult that their machine learning-based models precisely identify the minority classes.

There are also approaches based on rules. MacCartney and Manning (2009) have proposed a model of *natural logic*, which has focused on semantic containment and monotonicity. They also infer implicatives and factives based on *implication signatures* (Nairn et al., 2006) compositionally. But certainty is not considered in their approach. Saurí and Pustejovsky (2012) have proposed a rule-based method using information that can influence the factuality of events such as polarity particles, modality markers, and epistemic predicates. In their algorithm, factuality values of the event, consisting of certainty and polarity, are determined by the upper factuality values and rules, one by one, from the top of the dependency tree. Their model is suitable as it assumes the availability of a factuality lexicon and uses it to identify the factuality status of each subordinate event in a compositional manner from the factuality status of its superordinate event. So we adopt their model and adapt it to the Japanese language to discuss issues regarding lexical knowledge.

## 3 Japanese factuality analyzer

To discuss the problems about lexical knowledge, we construct a Japanese factuality analyzer based on the lexicon-based compositional approach proposed by Saurí and Pustejovsky (2012). Their analyzer is suitable for analyzing issues because it is based on the availability of a factuality-related simple lexicon and analogous lexicons for Japanese are also available. When we input a result of syntactic parsing to our factuality analyzer, it outputs the factuality of each event.

### 3.1 Factuality values

Saurí and Pustejovsky characterized a degree of event factuality as a pair of certainty (what is certain *vs* what is only possible) and polarity (positive *vs* negative). They divided the certainty axis into the values *certain* (CT), *probable* (PR), *possible* (PS) and *underspecified* (U), and the polarity axis into *positive* (+), *negative* (−) and *underspecified* (u). For example, an event “*de*” (*leave*) in (1a) is labeled with CT+. This means that it is certain that the event happened or will happen according to the author of the text. In the same way,

Table 1: Our Factuality values

certainty \ polarity	positive (+)	negative (-)
<i>certain</i> (CT)	fact (CT+)	counterfact (CT-)
<i>probable</i> (PR)	probable (PR+)	not probable (PR-)
<i>underspecified</i> (U)	unknown or uncommitted (U)	

an event “*kaet*” (*go home*) in (1b) is labeled with PR+ and “*hassei-suru*” (*occurrence*) in (1c) is labeled with CT-. We use Saurí and Pustejovsky’s factuality values; however, we make some changes to compensate for Japanese sentences.

The first is the distinction between PR and PS. In English, event factuality can be interpreted by specific expressions. For instance, PR is interpreted by *probable* and PS is interpreted by *possible*. However, in Japanese, it is not so straightforward to distinguish between PR and PS due to a diverse variety of modality expressions. Furthermore, PR and PS are minority classes. We therefore combine PR and PS into PR in order to focus on the distinction between certain and uncertain.

The second is underspecified values. Saurí and Pustejovsky used two underspecified values: the partially underspecified CTu and the fully underspecified Uu. For simplification, we do not distinguish two underspecified values. Instead we use U as the underspecified value.

Furthermore, in the present study, we start with focusing only on event factuality attributed to the author of the text. Analyzing factuality for other discourse participants is left for our future work.

We use Saurí and Pustejovsky’s factuality values except for these changes. In other words, we divide the certainty axis into the values *certain* (CT), *probable* (PR) and *underspecified* (U), and we also divide the polarity axis into *positive* (+) and *negative* (-). Table 1 shows factuality values by a combination of certainty and polarity.

### 3.2 Lexical knowledge

In Saurí and Pustejovsky’s model, the factuality is analyzed based on lexical knowledge, expressions (called *factuality markers*) that can influence the event factuality. For example, polarity particles of negation, such as the adverb *not*, switch the original polarity of its context, and particles of certainty, such as the auxiliary *may*, change the original certainty of its context. Saurí and Pustejovsky consider not only particles but also predicates. For instance, in the case of the expression *know that*, it presupposes that the event in *that*-clause is fac-

Table 2: Example entries of the dictionary of Japanese functional expressions

Sense Category	Expressions	Effects on Factuality
negation	<i>-nai</i> <i>-nu</i>	polarity: + → - - → +
speculation	<i>-daro-u</i> <i>-kamo-shire-nai</i>	certainty: CT → PR
question	<i>-ka</i> <i>-ka-na</i>	certainty: CT → U PR → U

Table 3: Example entries of the dictionary of Japanese clue expressions for extended modality

Expression	Tense of Embedded Event	Context Polarity	Factuality
<i>fusegu</i> ( <i>prevent</i> )	non-perfective	+	CT- CT+
<i>wasureru</i> ( <i>forget</i> )	non-perfective	+	CT-
	perfective	-	CT+
		+	CT+
		-	CT+

tual. Therefore, the predicate *know* is a factuality marker which changes the factuality of the event in *that*-clause into CT+.

Similarly, in Japanese, some expressions correspond to English factuality markers. We use the dictionary of Japanese functional expressions (Matsuyoshi et al., 2007) and the dictionary of Japanese clue expressions for extended modality (Eguchi et al., 2010) as factuality markers.

The dictionary of Japanese functional expressions is semantically categorized and contains a lot of functional expressions using a hierarchy with nine abstraction levels such as sense and grammatical function. This dictionary includes 341 direction words (16,711 expressions). We can use some categories as factuality markers. Table 2 shows example entries of this dictionary and corresponding effects on factuality. For instance, expressions categorized as speculation, such as “*-daro-u*” (*may*) seen in (1b), change the original certainty of its context. We use 5,345 expressions selected according to categories as factuality markers.

The dictionary of Japanese clue expressions for extended modality contains how predicates influence extended modality of surrounding events. This dictionary includes 8,122 predicates selected from Bunrui Goiho (National Institute for Japanese Language and Linguistics, 2004). These predicates also relate to the factuality. Therefore, we can use these predicates as factuality markers. Table 3 shows example entries of this dictionary and corresponding factuality. For example, the predicate “*fusei-da*” (*prevented*), seen in (1c), is regarded as the factuality marker that switches the polarity of the preceding event “*hassei-suru*” (*occurrence*).

Dependency tree	Factuality markers	Contextual factuality	Event factuality
		CT+ (initial value)	
知らない <i>shira-nai</i> (does not know)	-nai (not) polarity: + → -	CT-	<i>shira</i> (know): <u>CT-</u>
相手を <i>aite-wa</i> (the opponent)	<i>shira</i> (know) certainty: CT polarity: + → -	CT+	
ことを <i>koto-wo</i> (that)		CT+	
断念した <i>dannen-shi-ta</i> (had abandoned)	<i>dannen-shi</i> (abandon) polarity: + → -	CT+	<i>dannen-shi</i> (abandon): <u>CT+</u>
彼が <i>kare-ga</i> (he)		CT-	
出場を <i>shutsujou-wo</i> (the participation)		CT-	<i>shutsujou</i> (participation): <u>CT-</u>

Figure 1: Computing event factuality in (2)

### 3.3 Algorithm

The factuality analyzer determines an event factuality by propagating a pair of certainty and polarity along a dependency tree from the root of the sentence. The algorithm can reflect dependency between events by the propagation of the factuality. The algorithm determines the factuality of an event based on following components:

#### Predicates

The factuality is updated by predicates of its context.

#### Functional Expressions

The factuality is updated by functional expressions attached to the event.

#### Propagated Factuality

The factuality is determined based on the original factuality of the preceding event.

Figure 1 shows the analysis process when our algorithm is applied to (2). The input is the dependency tree of the sentence (2) (the left side of Figure 1) and the output is the factuality of each event (the right side of Figure 1).

- (2) 彼が出場を断念したことを相手は知らない。  
*kare-ga shutsujou-wo dannen-shi-ta-koto-wo aite-wa shira-nai.*  
 (The opponent does not know that he had abandoned the participation.)

First of all, the factuality at the top level is set to CT+ as initial value (by the naïve assumption), and

the factuality is propagated along a dependency tree from the root of the sentence. The process at each phrase consists of 3 steps.

As a first step, the analyzer updates the contextual factuality if the functional expression is found in the dictionary of Japanese functional expressions. For the first phrase “*shira-nai*” (does not know) in this example, the contextual factuality is updated to CT- by the negation “-nai” (not). As a second step, the factuality value is assigned to every found event. The factuality value CT- is assigned to the event “*shira*” (know) in the example. As a third step, the analyzer updates if the predicate is found in the dictionary of Japanese clue expressions for extended modality. In the example, the contextual factuality is updated to CT+ by the factive predicate “*shira*” (know). In referring to dictionaries in first and third steps, we adopt simple longest match for the surface. The third step needs to be performed after the second step due to the double nature of predicates, which are both event-denoting expressions and, at the same time, factuality markers.

Similarly, for the phrase “*dannen-shi-ta*” (had abandoned), the algorithm outputs CT+ as the factuality of the event “*dannen-shi*” (abandon), because of *Propagated Factuality* CT- (the factuality of the preceding event “*shira*” (know)), *Predicates* “*shira*” (know) (CT- → CT+) and *Functional Expressions* (empty for this case). The analyzer iterates the propagation and updates the con-

Table 4: Correspondence of *actuality* to factuality

certainty \ polarity	+	-
CT	certain+ certain- → certain+	certain- certain+ → certain-
PR	probable+ probable- → probable+	probable- probable+ → probable-
U	unknown	

textual factuality. As a result, CT- as the factuality of the event “*shira*” (*know*), CT+ as the factuality of the event “*dannen-shi*” (*abandon*), and CT- as the factuality of the event “*shutsujou*” (*participation*) are obtained.

## 4 Findings from empirical evaluation

### 4.1 Data and experimental setup

We apply our algorithm to 6,404 sentences on the Yahoo! Japan Q&A section for the Japanese corpus with extended modality (Matsuyoshi et al., 2010). These sentences are included in the *Balanced Corpus of Contemporary Written Japanese* (BCCWJ)<sup>1</sup>, and each event mention is labeled with extended modality (*source, time, conditional, primary modality type, actuality, evaluation, and focus*). *Actuality* denotes the degree of certainty and corresponds to our factuality. Table 2 shows the correspondence of *actuality* to our factuality.

In this experiment, we apply our algorithm to 11,395 event mentions, where *source* is “wr” (writer of the sentence). These event mentions are also selected by part-of-speech, such as verb and adjective. For the identification of the event mention, we give the gold data to the analyzer because we discuss only about lexical knowledge.

If the analyzer makes an error in regards to the factuality of an event, then this error will have an influence on the factuality of the next event, because the analyzer propagates the updated factuality to the next event. Our intent for this experiment is not to analyze this kind of error. Therefore, we use the gold label as *Propagated Factuality* in order to prevent the error propagation.

### 4.2 Discussion

We discuss issues about lexical knowledge through the error analysis of the analyzer based on lexical knowledge and compositionality. Our algorithm computes the event factuality based on *Predicates, Functional Expressions* and *Propagated Factuality*, but for matrix clauses, it determines the factuality based only on *Functional Expressions*. We expect issues to arise for func-

<sup>1</sup>[http://www.ninjal.ac.jp/corpus\\_center/bccwj/](http://www.ninjal.ac.jp/corpus_center/bccwj/)

Table 5: Accuracy for each case

	Matrix clauses	Subordinate clauses	Total
Correct	3,529	3,652	7,181
Wrong	693	3,521	4,214
Accuracy	0.836	0.509	0.630

Table 6: Confusion matrix for the certainty axis at matrix clauses

gold \ system	CT	PR	U	Total	Recall
CT	<b>2,478</b>	47	230	2,755	0.899
PR	145	<b>63</b>	50	258	0.244
U	151	11	<b>1,047</b>	1,209	0.866
Total	2,774	121	1,327	4,222	
Precision	0.893	0.521	0.789		

Table 7: Confusion matrix for the polarity axis at matrix clauses

gold \ system	+	-	Total	Recall
+	<b>2,374</b>	59	2,433	0.976
-	7	<b>293</b>	300	0.977
Total	2,381	352	2,733	
Precision	0.997	0.832		

Table 8: Confusion matrix for the certainty axis at subordinate clauses

gold \ system	CT	PR	U	Total	Recall
CT	<b>3,335</b>	330	1,997	5,662	0.589
PR	245	<b>104</b>	175	524	0.198
U	329	41	<b>617</b>	987	0.625
Total	3,909	475	2,789	7,173	
Precision	0.853	0.219	0.221		

Table 9: Confusion matrix for the certainty axis at subordinate clauses

gold \ system	+	-	Total	Recall
+	<b>3,224</b>	434	3,658	0.881
-	55	<b>301</b>	356	0.846
Total	3,279	735	4,014	
Precision	0.983	0.410		

tional expressions at matrix clauses. At subordinate clauses, on the other hand, we expect complex issues involving multiple components. We therefore analyze both the issues at matrix clauses and the issues at subordinate clauses, respectively.

Table 5 shows accuracy and Tables 6-9 show each confusion matrices for the certainty axis and the polarity axis for each case. These tables show that minority classes PR and U are difficult on the certainty axis. On the polarity axis, we obtain relatively high accuracy. Comparing matrix clauses to subordinate clauses, accuracy at subordinate clauses, which is based on some components, is lower than the accuracy at matrix clauses, which is based only on functional expressions. For each minority label (PR and U on the certainty axis, and - on the polarity axis), subordinate clauses have lower precision relative to matrix clauses. One reason for this is that we do not consider the scope of negation and speculation.

Table 10 shows the error type distribution. At

Table 10: Error type distribution

	Analyzed errors	Error type		Errors
Matrix clauses	108	functional expressions	semantic ambiguity	102
			insufficient coverage	4
		others		2
Subordinate clauses	1,041	functional expressions	semantic ambiguity	412
			insufficient coverage	16
		predicates	semantic ambiguity	4
			insufficient coverage	34
		scope		656

matrix clauses, the issue regarding functional expressions is found for 106 errors when analyzing 108 errors, and the rest of errors are due to an adverb and the parsing error. At subordinate clauses, we analyze 1,041 errors. Issues regarding the functional expressions (428 errors), predicates (38 errors), and the scope (656 errors) are found. Some errors are due to multiple issues. In the following paragraphs, we describe these issues in detail.

#### 4.2.1 Functional expressions

Out of the 106 errors for functional expressions, 53 false-positive errors regarding U were most common. Almost all of these errors are due to semantic ambiguity for functional expressions.

- (3) 知らないのも不思議ではないです。  
*shira-nai-no-mo fushigi-de-wa-nai-desu.*  
*(It is no wonder that he doesn't know.)*  
 (Gold: CT-, System output: U)

(3) is an example for semantic ambiguity of the functional expressions. Our analyzer refers to dictionaries by simple longest match. Therefore, the factuality of the event “*fushigi*” (*wonder*) is wrongly assigned as U because “*-de-wa*” is recognized as a recommendation (*how about*). In this context, the expression “*-de-wa*” is a part of inflection. So it has no special meaning.

As seen above, semantic ambiguity for functional expressions is a critical problem for Japanese factuality analysis. But disambiguation of Japanese functional expressions is not simple. Some previous work is engaged on this task, such as Tanaka et al. (2013). They construct MCN corpus for the disambiguation of expressions related to factuality. It is important to import this line of prior work to our analyzer.

Coverage for the dictionary of Japanese functional expressions also becomes a problem. However, the number of problems contains only 4 errors. We find that coverage for the dictionary of Japanese functional expressions is sufficient.

#### 4.2.2 Predicates

At subordinate clauses, 38 errors arise which are caused by predicate issues. 34 of the 38 errors are due to insufficient coverage for predicates and the other 4 errors are due to semantic ambiguity for predicates.

- (4) 正しいことを確認してください。  
*tadashii koto-wo kakunin-shi-te-kudasai.*  
*(Please check that it is correct.)*  
 (Gold: CT+, System output: U)

(4) is an example of insufficient coverage for predicates. In (4), our algorithm assigns U as the factuality of the event “*tadashii*” (*correct*) because U (the factuality of the event “*kakunin-shi*” (*check*), which is influenced by the request expression “*kudasai*” (*please*)) is propagated without any update. However, the predicate “*kakunin-shi*” (*check*) presupposes that the preceding context is factual, so it should be assigned CT+ as the factuality of the event “*tadashii*” (*correct*). This incorrect assignment occurs because that predicate does not exist in the dictionary of Japanese clue expressions for extended modality.

Out of 1,041 errors at subordinate clauses, 417 events are that predicates in the dictionary of Japanese clue expressions for extended modality are used. Only 4 errors, however, are due to semantic ambiguity for predicates. We therefore find that semantic ambiguity for predicates poses little problem. Furthermore, we focus on correct events by predicates. Out of the 1,128 correct instances in the area analyzed by the corpus, 351 are correct by predicates in the dictionary. In contrast to this, only 34 errors are due to insufficient coverage for predicates. For this reason, we find that insufficient coverage for predicates is a small issue.

#### 4.2.3 Scope

In Section 3, we described that our analyzer determines the event factuality based on three components: *Predicates*, *Functional Expressions*, and *Propagated Factuality*. However, we find that it

is crucial to determine boundaries whether the analyzer should propagate the factuality. In other words, it should resolve the scope of negation and speculation though the actual analyzer regards all embedded contexts as the scope. The errors due to the scope, in fact, are the majority of errors at subordinate clauses (656/1,041).

- (5) 少し郊外にでると音が聞き取れません。  
*sukoshi kougai-ni deru-to onsei-ga kikitore-mase-n.*  
*(I cannot hear the voice if I leave the suburbs.)*  
 (Gold: CT+, System output:  $\overline{\text{CT-}}$ )

Our algorithm wrongly assigns – as the polarity of the event “*deru*” (*leave*) in (5). This is because – (the polarity of the event “*kikitore*” (*hear*), which is influenced by the negation “*-n*” (*cannot*)) is propagated with no update. The negation “*-n*” (*cannot*) denies only the event “*kikitore*” (*hear*) but not the event “*deru*” (*leave*). As exemplified, the issue regarding the scope of negation and speculation is very crucial.

Of the 233 events where the analyzer outputs – as the polarity and the gold *Propagated Factuality* is –, 28 events are correct for the polarity, whereas 112 events are errors due to the scope. As shown, there are many cases where the analyzer should not propagate due to scope, and there are also many cases where the analyzer should propagate as –. We find that resolving the scope is a significant, but difficult challenge.

Next, we focus on the conjunction particles, such as “*-to*” in (5), as the key to detect scope in practice. Out of 656 errors due to the scope, the conjunction particle “*-to*” follows 126 events, “*-ga*” follows 78 events, “*-te*” follows 70 events, and so on. Therefore, when we detect scope in practice, we assume to use conjunctive particles as the key to determine propagation boundaries. In the next section, we investigate scope detection based on such expressions.

## 5 Lexicon-based scope detection

In the previous section, we found that detecting a scope is very crucial. In this section, we investigate the limitation of the lexical knowledge for a scope and identify the technical research issues more precisely through experiments for rule-based scope detection.

### 5.1 Related work for scope detection

In recent years, the detection of negation and speculation scopes is intensively being researched for English (Szarvas et al., 2008; Apostolova et al., 2011), such as Shared Task

in CoNLL-2010 (Farkas et al., 2010) and \*SEM 2012 (Morante and Blanco, 2012). For example, the BioScope corpus (Szarvas et al., 2008) is annotated with negation and modality expressions with their scope, and it is extensively used for resolution of the scope. However, studies for the detection for scope are insufficient for Japanese. Detection of scope in Japanese is a significant challenge, and will be highly beneficial for Japanese factuality analysis.

### 5.2 Knowledge-based scope detection

We take a rule-based scope detection approach to block propagating a contextual factuality. Before the first step on each phrase as described in Section 3, this approach blocks the propagation when the specific expressions are found in the event. The approach then assigns the contextual factuality as initial value CT+ and restarts the propagation. When such expressions are not found, the propagation is not blocked.

We used the terms shown in Table 11 to detect such expressions. When one of the terms appears at the end of an event, the event blocks the propagation. The terms are categorized by Minami (1974) according to the intensities of the constructing subordinate clauses: A is high, C is low and B is intermediate. These intensities would be used as a tendency of blocking the propagation. However, because there are some ambiguities such as “*~ ㇿ*” (*-te*) which belongs to all categories, we used all terms to detect scope and block the propagation.

### 5.3 Results

Table 12 shows the experimental results with/without lexical knowledge for scope. In the previous experiments as described in Section 4, in order to avoid the propagation error, we used gold contextual factuality. However, in our experiment, we focus on the propagation, so we do not use gold contextual factuality.

Table 12 shows that  $F_1$ -score increases 19.2% (0.112) by adding lexical knowledge. Focusing on each labels, our approach had no negative effect except recall of U. This means that our approach based on lexical knowledge works well, especially for minor labels. However, some errors still remain.

### 5.4 Remaining issues

We identify the remaining issues through the error analysis of the result. We focus on the

Table 11: Expressions to prevent propagating a contextual factuality

Category	Expressions
A	～ながら (-nagara), ～つつ (-tsutsu), ～て (-te), ～で (-de)
B	～て (-te), ～と (-to), ～ながら (-nagara), ～ので (-no-de), ～のに (-no-ni), ～ば (-ba), ～たら (-tara), ～なら (-nara), ～ても (-te-mo), ～て (-te), ～ず (-zu), ～ずに (-zu-ni), ～ないで (-nai-de)
C	～が (-ga), ～から (-kara), ～けれど (-keredo), ～けれども (-keredo-mo), ～けども (-kedo-mo), ～けど (-kedo), ～し (-shi), ～て (-te)

Table 12: Performance with/without lexical knowledge for scope

		CT+	PR+	PR-	CT-	U	Micro-Average	Macro-Average
The number of events		7,569	678	104	848	2,196	11,395	
With lexical knowledge	Precision	0.850	0.372	0.123	0.605	0.455	0.696	0.481
	Recall	0.753	0.178	0.067	0.672	0.697	0.696	0.474
	$F_1$	0.799	0.241	0.087	0.637	0.551	0.696	0.463
Without lexical knowledge	Precision	0.850	0.321	0.060	0.451	0.348	0.584	0.406
	Recall	0.584	0.156	0.048	0.542	0.756	0.584	0.417
	$F_1$	0.692	0.210	0.053	0.492	0.477	0.584	0.385

events which have a propagated factuality; in other words, it is not the last event of the sentence. In addition, the events whose propagated factuality is CT+ are also excluded from the analysis target, because when a CT+ is propagated to an event, even if the event blocks or doesn't block the CT+, the propagated factuality to the first step is CT+.

There are 1,739 events which satisfy the above conditions and we apply the block rule to 925 of them (i.e. some terms in Table 11 are found in the events). Table 13 shows the changes in the number of correct and incorrect results by adding the lexical knowledge. When using the block rule, 553 out of 925 incorrect events become correct. On the other hand, 100 of the correct events became incorrect. This suggests some ambiguities of expressions caused too much blocking.

- (6) a. 資格をうまく活かして働くことができなかった。  
shikaku-wo umaku ikashi-te hataraku koto-ga deki-nakat-ta.  
(I could not work by making best of my qualification.)
- b. 今は諸事象があって離婚できない。  
ima-wa shojijou-ga at-te rikon-deki-nai.  
(I cannot get a divorce because I have various reasons.)

For example, “～て” (-te) in (6a) causes blocking but in (6b) should not cause blocking.

Focusing on the coverage of the lexical knowledge, as described in Section 4, there are 656 errors due to the error of scope detection. 402 of them do not have CT+ as the propagated factuality and all of them should block the factuality propagation. However, only 229 of 402 blocked the propagation. This shows that the coverage of the lexical knowledge is still limited.

- (7) 半年前の点検では異常がみられなかった。  
hantoshi-mae-no tenken-de-wa ijou-ga mi-rare-nakat-

Table 13: Result changes by adding lexical knowledge

		with	
		correct	wrong
without	correct	49	100
	wrong	553	223

ta.  
(There are no defect in checking half a year ago.)

For example, “～では” (-de-wa) in (7) is not covered in this lexical knowledge.

## 6 Conclusion

We described Japanese factuality analysis, which is useful for information extraction and textual entailment recognition, among others. We discussed issues regarding lexical knowledge through error analysis by using a Japanese factuality analyzer based on lexical knowledge and compositionality. As a result, coverage of existing lexical resources is sufficient but issues regarding the semantic ambiguity of functional expressions and issues regarding scope were found. In particular, it was revealed that the problem regarding scope is most significant. We therefore performed an additional experiment with lexical knowledge for scope and discussed its helpfulness. However, the issue regarding scope includes the issue by profound meaning and context. Therefore, we consider that this issue is high-priority challenge.

In the future, we will address these challenges toward a high-performance Japanese factuality analyzer with other lexical knowledge and linguistic phenomena. Furthermore, we aim to construct a Japanese modality analyzer through the extension of the framework for factuality.

## Acknowledgement

This work was supported by MEXT KAKENHI Grant Number 23240018.

## References

- Emilia Apostolova, Noriko Tomuro, and Dina Demner-Fushman. 2011. Automatic extraction of lexico-syntactic patterns for detection of negation and speculation scopes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 283–287.
- Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational Linguistics*, 38(2):301–333.
- Megumi Eguchi, Suguru Matsuyoshi, Chitose Sao, Kentaro Inui, and Yuji Matsumoto. 2010. An analyzer of modality, actuality and valuation of events in japanese. In *Proceedings of the 16th Annual Meeting of Natural Language Processing (in Japanese)*, pages 852–855.
- Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 1–12.
- Kentaro Inui, Shuya Abe, Kazuo Hara, Hiraku Morita, Chitose Sao, Megumi Eguchi, Asuka Sumida, Koji Murakami, and Suguru Matsuyoshi. 2008. Experience Mining: Building a Large-Scale Database of Personal Experiences and Opinions from Web Documents. In *the 2008 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 314–321.
- Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eighth International Conference on Computational Semantics (IWCS-8)*.
- Suguru Matsuyoshi, Satoshi Sato, and Takehito Utsuro. 2007. A dictionary of japanese functional expressions with hierarchical organization. *Journal of Natural Language Processing (in Japanese)*, 14:123–146.
- Suguru Matsuyoshi, Megumi Eguchi, Chitose Sao, Koji Murakami, Kentaro Inui, and Yuji Matsumoto. 2010. Annotating event mentions in text with modality, focus, and source information. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1456–1463.
- Fujio Minami. 1974. *Structure of modern Japanese (in Japanese)*. Taishukan Shoten.
- Roser Morante and Eduardo Blanco. 2012. \*SEM 2012 shared task: Resolving the scope and focus of negation. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task*, pages 265–274.
- Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of Inference in Computational Semantics (ICoS-5)*.
- National Institute for Japanese Language and Linguistics. 2004. *Bunrui Goihyo (Word List by Semantic Principles)*. Dainippon-tosho.
- Roser Saurí and James Pustejovsky. 2009. FactBank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227–268.
- Roser Saurí and James Pustejovsky. 2012. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299.
- György Szarvas, Veronika Vincze, Richárd Farkas, and J. Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45.
- Ribeka Tanaka, Daisuke Bekki, and Ai Kawazoe. 2013. MCN corpus: The design and operation of the guidelines. In *Proceedings of the 19th Annual Meeting of Natural Language Processing (in Japanese)*, pages 77–80.



# A Hierarchical Semantics-Aware Distributional Similarity Scheme\*

Shuqi Sun<sup>1</sup>, Ke Sun<sup>2</sup>, Shiqi Zhao<sup>2</sup>, Haifeng Wang<sup>2</sup>, Muyun Yang<sup>1</sup>, and Sheng Li<sup>1</sup>

<sup>1</sup>Harbin Institute of Technology, Harbin, China

{sqsun, ymy}@mtlab.hit.edu.cn, lisheng@hit.edu.cn

<sup>2</sup>Baidu, Beijing, China

{sunke, zhaoshiqi, wanghaifeng}@baidu.com

## Abstract

The context type and similarity calculation are two essential features of a distributional similarity scheme (DSS). In this paper, we propose a hierarchical semantic-aware DSS that exploits semantic relation words as extra context information to guide the similarity calculation. First, we define and extract five types of semantic relations, and then develop relation-based similarities from the distributional similarities among the top-ranked relation words. Finally, we integrate various similarities using learning-to-rank technique. Experiments show that semantic relations are beneficial to predicting accurate similarity. On 6904 pairwise similarity comparisons, the predictive accuracy of our approach reaches 83.9%, which significantly outperforms the baseline approaches. We also conduct intrinsic analysis by varying the quality of semantic relations and the usage of individual similarities.

## 1 Introduction

Distributional similarity is an essential measure of the semantic relatedness between a pair of linguistic objects, including words, phrases, or even sentences. Confident distributional similar objects, are useful in various NLP applications, such as word sense disambiguation (Lin, 1997), lexical substitution (McCarthy and Navigli, 2009), paraphrase ranking (Dinu and Lapata, 2010), text classification (Baker and McCallum, 1998), etc. In this paper, we focus on the semantic similarity between words.

Well-known implementations of word-level distributional similarity scheme (DSS) mainly fall

---

This work was done when the first author was visiting Baidu.

into two categories according to the choice of the context: a) text-window based, and b) dependency path based. The former has the advantages of language-independence and computational efficiency, while the latter captures finer word-word relationships. However, both approaches focus on the usage aspect of words' meanings, which is only indirect indicator of the underlying semantic interactions.

Meanwhile, a great many successful efforts have been made to extract word-level semantic knowledge from the text, including synonyms / antonyms, sibling terms, hypernyms / hyponyms, holonyms / meronyms, etc. Despite of such deliberated studies, there are few considerations about how these semantics-oriented outcomes could contribute to the construction of DSS.

To realize the full potential of such semantic evidences, we propose a semantics-aware DSS by using semantic relation words to guide the similarity calculation. Our motivation is that words having similar semantic relational words, e.g. sibling terms or hypernyms, tend to be more semantically similar. The proposed DSS has a hierarchical layout, in which text-window based distributional similarity is first established from the corpus serving as the basis of the relation-specific similarities for 5 semantic relations. Finally, these similarities are linearly combined using learning-to-rank technique into a single measure, capturing both the context distributions and the semantic relations of the target words.

Our contribution is three-fold. First, by deriving semantic relation based similarities from distributional similarity, we develop a semantics-aware DSS in a hierarchical fashion. The DSS eventually fuses these similarities, and yields significant improvement over several baseline approaches. Second, our design of DSS relies solely on the same corpus where distributional similarity can be derived. It is adaptable to different languages, in

which distributional similarity and semantic relations are available. Third, our DSS's hierarchical nature allows us to individually replace each component with better implementations to adapt to specific applications or new languages.

## 2 Related Work

Distributional similarity (a.k.a. contextual similarity) has been elaborately studied to predict semantic similarity, and the type of the context is a main concern. A variety of context types have been proposed to capture the underlying semantic interactions between linguistic objects, including text-window based collocations (Rapp, 2003; Agirre et al., 2009), lexico-syntactic patterns (Turney, 2006; Baroni and Lenci, 2010), grammatical dependencies (Lin, 1998; Padó and Lapata, 2007; Thater et al., 2010), click-through data (Jain and Pennacchiotti, 2010), selectional preferences (Erk and Padó, 2008), synsets in thesaurus (Agirre et al., 2009), and latent topics (Dinu and Lapata, 2010). There are also researches that focus on distribution compositions (Mitchell and Lapata, 2008; Grefenstette et al., 2013) or context constrained similarity calculation (Erk and Padó, 2008).

Extracting sibling or hierarchical semantic relations from corpora forms a different track of research, in which exist ample efforts. Most of them make use of hand-crafted or automatically bootstrapped patterns. Various types of patterns have been tried out, including plain texts (Hearst, 1992; Paşca, 2004), semi-structured HTML tags (Shinzato and Torisawa, 2007), or their combinations (Shi et al., 2010). Bootstrapping approach is shown useful given a number of seeds, which could be either relation instances (Snow et al., 2004; Pantel and Pennacchiotti, 2006), or initial patterns (Pantel et al., 2004). To improve the quality of raw extraction results, some studies also resort to optimizing one relation using other relations (Zhang et al., 2011; Kozareva et al., 2011) or using distributional similarity (Shi et al., 2010).

Despite the great progress made in the field of semantic relation extraction, few studies explicitly use semantic relations to guide the similarity calculation. In this paper, we use instance-pattern iteration on a massive corpus to populate semantic relation instances, and derive relation-specific similarities on top of text-window based distributional similarity. Indeed, previous studies did resort to a closed set of lexical patterns that indicate

sibling / hypernym / hyponym relations (Baroni and Lenci, 2010; Bansal and Klein, 2012), concept properties (Baroni et al., 2010), and attribute information (Baroni and Lenci, 2010). Compared with these studies, our approach systematically exploits specific semantic relations instead of counting co-occurrence under surface patterns. We also develop a hierarchical similarity fusion architecture, rather than blending the heterogeneous evidences in a single distribution vector (Baroni and Lenci, 2010). It is also notable that sibling term extraction, in which various semantic evidences (e.g. hypernyms) also help, is not in the track of our study. Sibling term extraction focuses on words sharing the same super concept, and does not quantify the pair-wise similarity between them. Nevertheless, sibling terms work fine as an evidence of semantic similarity, as shown in our experiments.

Machine learning based integration of multiple evidences are shown useful in semantic class construction and semantic similarity calculation. Pennacchiotti and Pantel (2009) use gradient boosting decision tree to combine evidences from Web page, query log, Web tablet, and Wikipedia to populate instances of *Actors*, *Athletes*, and *Musicians*. There are also studies that combine distribution and pattern information in lexical entailment (Mirkin et al., 2006) and word clustering (Kaji and Kitsuregawa, 2008). Close to our work, Agirre et al. (2009) train SVM classification models to combine individual similarities derived from dependency path, text-window, and WordNet synsets. The synsets are highly accurate in representing words' meanings. However, the size of the thesaurus is limited, and not equally available in different languages. Although Agirre et al. (2009) tried machine translation techniques to tackle with this issue, abundant named entities and translation errors in the Web corpus still challenge the performance of their approach.

## 3 Hierarchical Semantics-aware DSS

Our proposed DSS has a four-layer structure, as shown in Figure 1. The bottom layer is the *corpus layer* where a massive Web page repository is pre-processed. Upwards, we build a *distribution layer* to obtain basic text-window based distributional similarity between any pair of target words. The distribution layer provides a distributional similarity database upon which a *semantics layer* takes effect. At this layer we adopt an extraction system

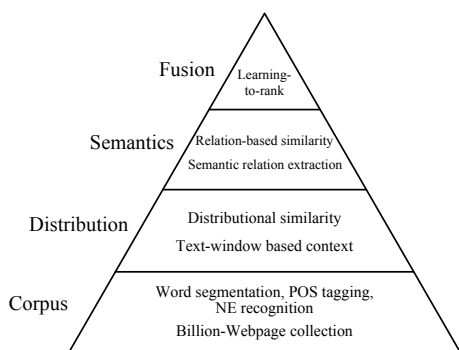


Figure 1: Hierarchical approach to semantics-aware DSS.

that iteratively populates instances for 5 types of semantic relations. Then, for each type of semantic relation, we develop a relation-specific similarity measure. Finally at the *fusion layer*, the similarities at the distribution and semantics layers are integrated linearly using learning-to-rank.

### 3.1 Corpus Layer

The corpus we work on is a repository of Chinese Web pages collected in 2011. It contains 1.1 billion pages ( $5.8 \times 10^{11}$  words in total), and takes 4.7TB of storage. All pages are de-tagged, leaving only their titles and textual content, and then segmented with a word segmentor based on dictionaries plus conditional random field (CRF) models. The segmentor is efficiently implemented, and is able to process 40K words per second with an accuracy around 98%. Based on the segmentation results, two more steps of pre-processing: POS tagging and named entity (NE) recognition are also performed.

### 3.2 Distribution Layer

The most widely studied types of context for distributional similarity are text-window based context and grammatical context. The construction of the latter requires syntactic or dependency parsing, which is highly language-dependant and may be extremely time-consuming on large corpora.

Therefore, at the distribution layer, we build the distributional similarity database using simple text-window based co-occurrences. Two different lengths of the text-window are experimented: 3 words and 6 words. The window slides one word per step from the beginning of each sentence to the end. Thus, the 3-length and 6-length windows capture two and five words at most on each side of the target word respectively. For each pair of words  $w$  and  $w'$ , four association measures are

tried out, covering a range of common practices, including (1) raw number of co-occurrence, (2) point mutual information (PMI), (3) Jaccard index, and (4) local mutual information (LMI) (Evert, 2008). To reduce the amount of computation, we preserve top 5000 context words for each target word. Processing the whole corpus yields  $\sim 6.5$  million unique target words<sup>1</sup>. For any pair of words  $w$  and  $w'$  in the vocabulary, the distribution layer provides it a cosine similarity between their context distributions, denoted by  $ds(w, w')$ .

### 3.3 Semantics Layer

We deem that semantic relationship is a more direct clue of a word’s *meaning* than either its text-window co-occurrences or syntactic dependencies. Therefore, we introduce a semantics layer upon the distribution layer to exploit semantic relations. Specifically, we adopt an extraction system to populate semantic relation instances, and then derive relation-based similarities from the system’s output.

#### 3.3.1 Relation Extraction

Here, we present a fully-featured, yet lightweight semantic relation extraction system that is capable to conduct in-depth mining in massive corpora. The system follows the line of instance-pattern iteration on a massive corpus, and can be substituted by any implementation of such fashion.

We define and extract five types of semantic relations, as listed in Table 1. The extraction starts in the first iteration with a number of seed relation instances. A relation instance is defined as a triple  $(w, r, w')$ , which means words  $w$  and  $w'$  have the relation  $r$ .

$r$	Relation	Description
$r_1$	$w [Sibling]_{is} w'$	$w, w'$ are sibling terms
$r_2$	$w [Hyponym]_{is} w'$	$w'$ “is a” $w$ $w'$ is a “Instance-of” $w$
$r_3$	$w [Hypernym]_{is} w'$	$w$ “is a” $w'$ $w$ is a “Instance-of” $w'$
$r_4$	$w [Meronym]_{is} w'$	$w'$ is a “Part-of” $w$ $w'$ is a “Member-of” $w$ $w'$ is a “Substance-of” $w$
$r_5$	$w [Holonym]_{is} w'$	$w$ is a “Part-of” $w'$ $w$ is a “Member-of” $w'$ $w$ is a “Substance-of” $w'$

Table 1: List of semantic relations.

In a nutshell, during a full iteration, the system

<sup>1</sup>This is much larger than the number of typical Chinese words, which is mainly caused by the huge amount of NEs in the Web corpus, plus typos and word segmentation errors.

first uses seed instances to populate initial patterns that match them from the corpus, and assigns a unique semantic relation to each pattern according to the seeds it matches. Then, the system uses these patterns to extract new word pairs, and assigns relations to the word pairs according to the patterns that extracted them. The whole procedure is described in detail as the following four steps:

**(1) Pattern initialization** Find all sentences that contain the words  $w$  and  $w'$  in any seed relation instance  $(w, r, w')$ . Each sentence is then split into *prefix*, *infix*, and *suffix* by  $w$  and  $w'$ . The total length limit of these three parts is 10 words. Within this limitation, the system exhaustively enumerates all possible *prefixes* and *suffixes* (the *infix* remains unchanged). For example, in the sentence “A B  $w$  C  $w'$  D E”, the *prefixes* can be {‘A’, ‘B’, ‘A B’, ‘ ’} and the *suffixes* can be {‘D’, ‘E’, ‘D E’, ‘ ’}. Then, each combination of “*prefix SLOT<sub>1</sub> infix SLOT<sub>2</sub> suffix*” forms the word level of a pattern. Plus the POS and NE tags, a pattern finally contains three levels of information. In addition, to increase the recall of the patterns, named entities at the word level are replaced by their NE tags. This means at these positions, the pattern would match an arbitrary word as long as the word’s NE tags are matched. For instance, say the following sentence matches a seed (苹果(Apple), [*Sibling*]<sub>is</sub>, 三星(Samsung)):

近日, [苹果]和[三星]在美国进行了专利诉讼。  
(Recently, [Apple] and [Samsung] conducted patent litigation in the U.S.)

A pattern derived from this sentence could be:

Recently {*SLOT<sub>1</sub>*} and {*SLOT<sub>1</sub>*} in U.S.  
Word: 近日 {*SLOT<sub>1</sub>*} 和 {*SLOT<sub>1</sub>*} 在 LOC  
POS: t nz c nz p ns  
NE: NOR BRD NOR BRD NOR LOC

where BRD, LOC stand for brand and location, and NOR means the word is not a NE.

**(2) Pattern-relation mapping** For pattern  $p$ , consider all seed instances  $(w, r, w')$  it matched. For each relation  $r$  in those seeds, count it w.r.t.  $p$ . Then,  $r$  is scored by *tfidf*, where *tf* is  $r$ ’s count, and *df* is the number of relations that have a non-zero count. Finally, map  $p$  to the relation  $r_{max}$  with the highest score  $s_{max}$ , and assign  $s_{max}$  to  $p$  as its score.

**(3) Instance extraction** For each semantic relation  $r$ , extract word pairs using top scored 1,000 patterns that are mapped to  $r$ . For each sentence, if it matches a pattern  $p$ ’s word sequence and meets

the POS / NE tag constraints at  $SLOT_1$  and  $SLOT_2$  in  $p$ , then the words falling into the two slots are extracted.

Note that different patterns (even those with different relations) may extract the same word pair. To determine the final relation of a word pair  $\langle w, w' \rangle$ , the system traverses all the patterns that can extract it. The patterns are then grouped by the relations they map to. Within each group, the patterns’ scores are added up. The relation  $r^*$  whose group has the highest sum score  $S$  is selected. Then, with  $r^*$ , a new instance  $(w, r^*, w')$  is generated, with  $S$  as its weight. The system will also generate a reversed instance  $(w', r^{*-1}, w)$ . E.g., for (Intel, [*Sibling*]<sub>is</sub> AMD) and (Intel, [*Hypernym*]<sub>is</sub> Company), the system also generates (AMD, [*Sibling*]<sub>is</sub> Intel) and (Company, [*Hypernym*]<sub>is</sub> Intel) respectively.

**(4) New seed generation** Add the top-weighted relation instances obtained in step (3) into the seed set. In the current setting, each relation’s top-500 weighted instances are added as new seeds in each iteration.

In practice, the seed set used is small ontology of totally 222k instances of the five relations. The system produces 27M instances after 2 iterations, which are used in our experiments. For each word  $w$ , we define its relation words as the words appearing in the slot “ $w$  [*relation*]<sub>is</sub> □”. E.g.,  $w$ ’s [*Hypernym*]<sub>is</sub> relation words are its hypernyms.

### 3.3.2 Similarity

Recall that at the distribution layer, each pair of words has a distributional similarity, denoted by  $ds(\cdot, \cdot)$ . On top of this, we individually develop a relation-based similarity  $rs_i(\cdot, \cdot)$  for each semantic relation  $r_i \in \{r_1, r_2, \dots, r_5\}$ . For two words  $w$  and  $w'$ ,  $rs_i(w, w')$  is defined as the average of non-zero distributional similarities between their top- $N$  (at most) relation words under  $r_i$  (denoted by  $r_i^N(\cdot)$ ):

$$rs_i(w, w') = \frac{1}{|\{(u, v) | ds(u, v) > 0\}|} \sum_{\substack{u \in r_i^N(w) \\ v \in r_i^N(w')}} ds(u, v) \quad (1)$$

In our experiments, we universally set  $N$  to 10.

One alternate practice is to directly calculate the traditional cosine similarity between the relation word distributions of  $w$  and  $w'$ . We do not take this approach because such manner suffers from data sparseness. In particular, sometimes the relation words of  $w$  and  $w'$  are quite similar, but none

of them are shared by  $w$  and  $w'$  (this means the cosine similarity will be 0). For instance:

- *hand* and *head* may not share any meronym, e.g. *hand* only has meronym *finger* while *head* has *eye*, *nose*, ...;
- *Carmel* (a small city in IN, U.S.) may only have the hypernym  $\{small\ city\}$  while *New York* may have  $\{city, big\ city\}$ .

In our approach, owing to the non-zero  $ds(\cdot, \cdot)$  between *finger* / *eye*, or between *small city* / *big city*, the two pairs of words will have positive relation-based similarities.

### 3.4 Fusion Layer: Learning-to-rank

Eventually, we fuse  $ds(\cdot, \cdot)$  and  $rs_i(\cdot, \cdot)$  together to get the final similarity prediction of each pair of words. We choose a straightforward manner by linearly combining  $ds(\cdot, \cdot)$  and  $rs_i(\cdot, \cdot)$ :

$$FUSE(w, w') = \alpha_d \cdot ds(w, w') + \sum_i \alpha_i \cdot rs_i(w, w') \quad (2)$$

Note that the relation-based similarities are built upon  $ds(\cdot, \cdot)$  (Eq. 1), so FUSE is essentially a hierarchical combination of  $ds(\cdot, \cdot)$  guided by semantic relations. Linear combination is simple, but turns out to be effective through the experiments. More elaborated fusion method may be invested in future studies.

To get the weights  $\alpha_d$  and  $\alpha_i$ , we adopt pair-wise learning-to-rank technique rather than regression. This is because it is difficult to assign an absolute score of semantic similarity to a pair of words, especially when seasoned linguists are not available. On the other hand, given two word pairs  $\langle A, B \rangle$  and  $\langle A, C \rangle$ , it is relatively easier to tell whether  $\langle A, B \rangle$  is more similar than  $\langle A, C \rangle$ , or vice versa.

We use the ranking option of SVM<sup>light</sup> v6.02 (Joachims, 1999) with linear kernel to optimize the weights against human judgements. The goal of the learning process is to minimize the number of wrong pair-wise similarity comparisons. In the testing phase, the model assigns to each testing sample a real-value prediction, which is exactly a linear fusion of the corresponding sub-similarities. As for the technical details in SVM<sup>light</sup>, each word pair  $\langle X, Y \rangle$  yields a sample. If the human judgement suggests  $\langle A, B \rangle$  is more similar than  $\langle A, C \rangle$ , then the corresponding samples will be assigned to an unique sample group, with the target values 1 and 0 respectively. If a sub-similarity

value does not exist due to out-of-vocabulary issue, the corresponding feature is set to “missing”.

The judgement is obtained from a Chinese thesaurus (HIT-SCIR, 2006), containing 77,458 words that are manually grouped according to a five-level category hierarchy. Words grouped together at the lowest(fifth) level include both synonyms (e.g. *sea* / *ocean*) and comparable terms (e.g. *Germany* / *France*). The lower the level two words appear in together, the more semantically related they are. We directly use this clue to determine the semantic similarity between words. For instance, words that appear together in a level-3 category but not in any level-4 category have a similarity of 3. Words do not appear together in any category have a zero-similarity.

After a pilot study, we found that words with similarities 0-2 are indistinguishably dissimilar. So we merged these similarity levels together as zero-similarity. Moreover, to further distinguish semantically-similar and comparable words, we set the similarity between synonyms to 6 instead of 5 for comparable terms. Finally, we got five similarity levels: 0, 3, 4, 5, and 6. To make the experiment manageable, we randomly sample 200 nouns from the thesaurus and extract their similar words at every level, and arrange them as a serial of similarity judgements like  $sim(w, w') > sim(w, w'')$ . The whole dataset contains 2,204 words and 6,904 judgements. To avoid the randomness in data, we adopt five-fold cross-validation on it. In each fold, we use 3 parts of the data to train the model, and tune / test it using the other two parts.

## 4 Evaluation

### 4.1 Experiment Settings

We compare our fused similarity with three baselines. The first one is classical text-window based distributional similarity  $ds(\cdot, \cdot)$ . The other two baseline approaches are listed as follows:

*Lin’s similarity* (Lin, 1998) (LIN98). LIN98 combines PMI values from different distributions linearly. The formula uses dependency paths, and we extend it to semantic relations extracted as in Section 3.3.1. As a by-product in the extraction phase, words’ co-occurrence counts under each semantic relation are acquired to compute the PMI values. The text-window based distribution is also included in the combination.

*Joint cosine similarity* (JCS). There is also previous work that uses pattern-constrained context

information as extra clue of semantic similarity (Baroni and Lenci, 2010). Different from LIN98, words’ co-occurrence counts under each semantic relation are replaced by the number of patterns that extract them (Baroni and Lenci, 2010). Here, the text-window and the relation based distributions are mingled into a single distribution, and cosine similarity is obtained. Baroni and Lenci (2010) uses LMI in relation based distributions, but we found PMI achieves better performance.

The text-window distribution used in both LIN98 and JCS is based on 3-length window and PMI, since this configuration shows the best performance in our experiments. For the relation based distribution in LIN98 and JCS, we initially use the whole (noisy) relation extraction result, and make further analysis by varying the amount (and quality) of the relation instances. LIN98 and JCS also generate integrated similarities based on multiple evidences, we will compare their effectiveness with our approach.

Two evaluations metrics are used:

*Accuracy of comparison (Acc.)*. We say a system makes a correct comparison if it returns  $S(A, B) > S(A, C)$  that coincides with human judgement. The overall accuracy is defined as the percentage of correct comparisons over the whole dataset.

*Spearman’s  $\rho$* . For each word pair  $\langle A, B \rangle$ , we count the number of word pairs  $\langle A, x \rangle$  that are judged less similar than  $\langle A, B \rangle$ , and use it as an absolute score of similarity between  $A$  and  $B$ . This allows us to compare similarity predictions with such scores globally, and get the  $\rho$  coefficient.

To get meaningful conclusions, we use approximate randomization (Noreen, 1989) to test the significance of *Acc.* comparison, and Steiger’s Z-test (Steiger, 1980) for Spearman’s  $\rho$  comparison.

## 4.2 $ds(\cdot, \cdot)$ Configurations

Distributional similarity  $ds(\cdot, \cdot)$  is an important baseline. Moreover, by substituting Eq. 1 into Eq. 2, one will find that  $ds(\cdot, \cdot)$  is also the basic building block of the fused similarity. With the multiple choices of the text-window lengths (3 and 6) and association measures (Raw co-occurrence, PMI, Jaccard, and LMI) listed in subsection 3.2, we now try to find out an optimal configuration of  $ds(\cdot, \cdot)$ . The results are obtained based on the whole dataset, as shown in Table 2. For the  $ds(\cdot, \cdot)$  configurations (the first 8 rows), the subscripts are

the text-window’s length and the superscripts are the association measures used. Performance of LIN98 and JCS is also included (rows 9~10).

	<i>Acc.</i>	$\rho$
$ds_{3wd}^{Raw\ cooc}(\cdot, \cdot)$	77.0	0.458
$ds_{6wd}^{Raw\ cooc}(\cdot, \cdot)$	75.2	0.427
$ds_{3wd}^{PMI}(\cdot, \cdot)$	<b>80.8</b>	0.522
$ds_{6wd}^{PMI}(\cdot, \cdot)$	77.4	0.438
$ds_{3wd}^{Jac.}(\cdot, \cdot)$	80.1	0.527
$ds_{6wd}^{Jac.}(\cdot, \cdot)$	79.0	0.501
$ds_{3wd}^{LMI}(\cdot, \cdot)$	80.0	<b>0.544</b>
$ds_{6wd}^{LMI}(\cdot, \cdot)$	78.2	0.497
LIN98	79.4	0.496
JCS	<b>82.2</b>	<b>0.553</b>

Table 2: Performance of  $ds(\cdot, \cdot)$ , LIN98, and JCS

In both *Acc.* and  $\rho$ ,  $ds(\cdot, \cdot)$  with the 3-length window significantly ( $p < 0.01$ ) outperforms that with the 6-length window, except when using Jaccard ( $p = 0.12$  for *Acc.*). Although the window length is easy to choose, it remains unclear which association measure is the most appropriate. With the 3-length window, the performance of PMI, Jaccard, and LMI are comparable. Thus, we will have to try out all PMI, LMI and Jaccard in the fusion phase.

As for the other two baseline approaches, JCS significantly outperforms all  $ds(\cdot, \cdot)$  configurations in both *Acc.* ( $p < 0.05$ ) and  $\rho$  ( $p < 0.07$ ) as shown in bold font, but LIN98 does not.

## 4.3 Similarity Fusion

In similarity fusion (Eq. 2), for the sake of conciseness, we use the same  $ds(\cdot, \cdot)$  configuration to compute both the distributional similarity and the relation-based similarities (Eq. 1). The *Acc.* and  $\rho$  of the fused similarity (denoted by FUSE) using different  $ds(\cdot, \cdot)$  configurations are shown in Table 3. Recall that because of the indistinguishable *Acc.*, three configurations need to be examined:  $ds_{3wd}^{PMI}(\cdot, \cdot)$ ,  $ds_{3wd}^{LMI}(\cdot, \cdot)$ , and  $ds_{3wd}^{Jac.}(\cdot, \cdot)$ .

	$ds(\cdot, \cdot)$ configuration used		
	$ds_{3wd}^{PMI}(\cdot, \cdot)$	$ds_{3wd}^{LMI}(\cdot, \cdot)$	$ds_{3wd}^{Jac.}(\cdot, \cdot)$
<i>Acc.</i>	<b>83.9</b>	81.9	78.9
$\rho$	<b>0.591</b>	0.558	0.500

Table 3: Performance of our proposed fused similarity (FUSE) using different  $ds(\cdot, \cdot)$  configurations.

In both *Acc.* and  $\rho$ ,  $ds_{3wd}^{PMI}(\cdot, \cdot)$  based FUSE has

significantly ( $p < 0.005$ ) superior performance (shown in bold font). In both metrics, it also significantly ( $p < 0.01$ ) outperforms all baseline approaches, including all  $ds(\cdot, \cdot)$  configurations, LIN98, and JCS. The results suggest that on our dataset, the most suitable  $ds(\cdot, \cdot)$  to use in FUSE is  $ds_{3wd}^{PMI}(\cdot, \cdot)$ , which achieves 83.9% accuracy in predicting whether a word pair  $\langle A, B \rangle$  is more similar than  $\langle A, C \rangle$ .

As a global comparison, we have the following performance rankings:

$$\begin{aligned} Acc. : & \text{LIN98} <_{0.05} ds_{3wd}^{PMI}(\cdot, \cdot) <_{0.05} \text{JCS} <_{0.01} \text{FUSE} \\ \rho : & \text{LIN98} <_{0.01} ds_{3wd}^{LMI}(\cdot, \cdot) <_{0.01} \text{JCS} <_{0.01} \text{FUSE} \end{aligned}$$

where the subscripts show the significance level. JCS outperforms the best  $ds(\cdot, \cdot)$  configurations in both  $Acc.$  and  $\rho$ , confirming the contribution of the semantic evidences obtained by the in-depth mining in the corpus. Moreover, FUSE achieves even better performance, showing the effectiveness of the design of relation-based similarity (Eq. 1) and the linear combination mechanism (Eq. 2).

Ideally, an effective fusion should have worked for all  $ds(\cdot, \cdot)$  configurations. However, FUSE using  $ds_{3wd}^{Jac}(\cdot, \cdot)$  yields bad performance. Through intrinsic analysis we found that  $ds_{3wd}^{Jac}(\cdot, \cdot)$  is more sensitive to the noise in the relation data than  $ds_{3wd}^{PMI}(\cdot, \cdot)$ .

#### 4.4 Quality of Semantic Relations

Initially, we use all of the extracted relation instances in the experiments without threshold based filtering. Without doubt, there is much noise in the bottom of the extraction results. Through controlling the weight threshold of the relation instances, we now shrink the global extraction results to top  $\sim 5\%$ ,  $\sim 10\%$ ,  $\sim 30\%$ , and  $\sim 60\%$  subsets to see how their quality and coverage change, and how they affect the performance of FUSE, LIN98, and JCS.

FUSE uses top 10 relation words to calculate the relation-based similarity. Thus, instead of examining the global extraction results, we focus on the top 10 relation words of the target words in our dataset, because FUSE’s performance is our main concern.

The full evaluation is expensive. There are totally 2,204 target words in the dataset, involving 70,000 relation words. So we randomly sample 200 words from the 2,204 words, and evaluate the accuracy of their relation words by varying the amount of the global extraction results. The results are summarized in Table 4. While the amount

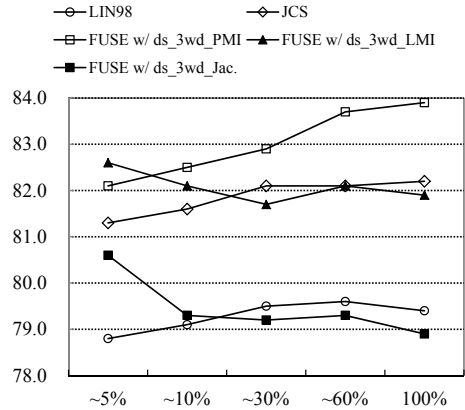


Figure 2: Accuracy of LIN98, JCS, and FUSE varying the amount of the global extraction results.

of global extraction results shrink, low-weight relation words are gradually removed, and the coverage of the relation words decreases. Only the top  $\sim 5\%$  results have acceptable accuracies. Intrinsic study shows that holonyms and meronyms concentrate to location names due to the bias in the seeds. This causes a significantly low quality and coverage for these two relations.

The low-quality extraction results pose an austere challenge. Here, we re-examine LIN98, JCS, and FUSE (5-fold CV using  $ds_{3wd}^{PMI}(\cdot, \cdot)$ ,  $ds_{3wd}^{LMI}(\cdot, \cdot)$ , or  $ds_{3wd}^{Jac}(\cdot, \cdot)$ ) based on the four subsets of the global extraction results, and assemble the performance figures in Figure 2. For the sake of space limit, we only include the  $Acc.$  results. Spearman’s  $\rho$  shows a similar trend.

The results further confirms the ranking listed in subsection 4.3. A common finding is that the bottom 40% of the global extraction results are hardly useful. FUSE based on  $ds_{3wd}^{PMI}(\cdot, \cdot)$  handles the noise in the data quite well. FUSE based on  $ds_{3wd}^{LMI}(\cdot, \cdot)$ , or  $ds_{3wd}^{Jac}(\cdot, \cdot)$  seems partial to high-quality data. Similar to  $ds_{3wd}^{PMI}(\cdot, \cdot)$  based FUSE, performance of LIN98 and JCS drops while shrinking the number of relation instances. This indicates they prefer recall to precision of the extraction results.

#### 4.5 Feature Analysis

We have shown that the fusion of  $ds_{3wd}^{PMI}(\cdot, \cdot)$  and  $rs_i(\cdot, \cdot)$  shows superior performance, yet each sub-similarity’s contribution remains unclear. Using 100% global extraction results, we incrementally add relation-based similarities to  $ds_{3wd}^{PMI}(\cdot, \cdot)$ , and report the fusion’s cross-validation performance in Table 5. The order of addition is coincide to the quality of the relations (see Table 4).

	Relation words	~5%		~10%		~30%		~60%		100%	
		# wd.	Acc.	# wd.	Acc.	# wd.	Acc.	# wd.	Acc.	# wd.	Acc.
$r_1$	$w$ [Sibling] <sub>is</sub> $\sqcup$	1,115	96.6	1,325	88.3	1,532	80.9	1,615	78.7	1,648	77.9
$r_2$	$w$ [Hyponym] <sub>is</sub> $\sqcup$	263	77.6	627	38.8	1,018	27.8	1,227	23.9	1,378	21.8
$r_3$	$w$ [Hypernym] <sub>is</sub> $\sqcup$	608	73.8	1,059	52.0	1,376	44.6	1,488	42.7	1,561	40.9
$r_4$	$w$ [Meronym] <sub>is</sub> $\sqcup$	266	41.7	416	29.6	586	22.7	741	19.4	972	15.1
$r_5$	$w$ [Holonym] <sub>is</sub> $\sqcup$	141	55.3	161	50.9	197	42.6	388	23.2	462	20.1

Table 4: Quantity and quality analysis of the 200 sampled words’ relation words.

Feature set	Acc.	$\rho$
$ds_{3wd}^{PMI}$	80.8	0.522
$ds_{3wd}^{PMI} + rs_1$	83.1	0.559
$ds_{3wd}^{PMI} + rs_1 + rs_3$	83.4	0.587
$ds_{3wd}^{PMI} + rs_1 + rs_3 + rs_2$	83.6	0.587
$ds_{3wd}^{PMI} + rs_1 + rs_3 + rs_2 + rs_5$	83.7	0.590
$ds_{3wd}^{PMI} + rs_{1\sim 5}$	83.9	0.591
$ds_{3wd}^{PMI} + rs_2$	81.3	0.522
$ds_{3wd}^{PMI} + rs_3$	82.3	0.574
$ds_{3wd}^{PMI} + rs_4$	81.2	0.532
$ds_{3wd}^{PMI} + rs_5$	81.1	0.550

Table 5: FUSE’s performance on sub feature sets.

Unsurprisingly,  $rs_1(\cdot, \cdot)$ , i.e. [Sibling]<sub>is</sub> based similarity is the most effective, owing to its high quality.  $rs_3(\cdot, \cdot)$  ([Hypernym]<sub>is</sub>) dominates the rest of the performance improvement, and adding it alone to  $ds_{3wd}^{PMI}(\cdot, \cdot)$  also largely improves the performance. It is reasonable since comparing the sibling terms or hypernyms (i.e. “what is it”) are natural ways to compare words’ meanings. Though “masked” by [Sibling]<sub>is</sub> and [Hypernym]<sub>is</sub>, other relations also show their contribution (yet small) when added to  $ds_{3wd}^{PMI}(\cdot, \cdot)$  alone.  $rs_2(\cdot, \cdot)$  ([Hyponym]<sub>is</sub>) is an exception, and its weight is also negative in the trained models. This indicates that hyponyms may not be an adequate evidence for semantic similarity.

Given the bad quality of [Meronym]<sub>is</sub> and [Holonym]<sub>is</sub> relations, their effectiveness seems bizarre. In fact, though a great number of relation words are not correct, they can be considered as special context words. Owing to the design of the relation-based similarity (Eq. 1), the distributional similarities of those words still contribute to the target words’ similarity calculation. This finding allows us to relax the quality restriction of semantic relation extraction. Our hierarchical approach to semantics-aware distributional similarity would work on the basis of noisy relation databases.

## 5 Conclusion

In this paper, we propose a hierarchical semantics-aware distributional similarity scheme (DSS). We introduce a semantic layer over the classical dis-

tribution layer by employing a semantic relation extraction system and a mechanism that computes words’ relation-specific similarities based on simple distributional similarity. Finally, the fusion of the distributional and relation-based similarities is completed by learning-to-rank.

Experiments show that the in-depth mining in the corpus provides effective evidences for semantic similarity. On our dataset, the fused similarity significantly improves distributional similarity, and also outperforms the baseline approaches that blend the heterogeneous evidences in a single vector. Additionally, intrinsic analysis shows that [Sibling]<sub>is</sub> and [Hypernym]<sub>is</sub> relations are the most effective semantic clues.

In future studies, we will experiment on more elaborated combination similarity fusion mechanisms other than linear combination. We will also explore more types of semantic evidences, e.g. synonym, antonym, semantic attribute, or thematic relations such as agent / patient relations.

## Acknowledgments

This work was supported by (1) the National High Technology Research and Development Program of China (863 Program, No. 2011AA01A207), (2) the Natural Science Foundation of China (No. 61272384 & 61105072), (3) China Postdoctoral Science Foundation (No. 2012M510220), and (4) Beijing Postdoctoral Research Foundation (No. 2012ZZ-99).

## References

- Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL ’09*, pages 19–27.
- L. Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In *SIGIR ’98*, pages 96–103.
- Mohit Bansal and Dan Klein. 2012. Coreference semantics from web features. In *ACL ’12*, pages 389–398.



- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721.
- Marco Baroni, Brian Murphy, Eduard Barbu, and Massimo Poesio. 2010. Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34(2):222–254.
- Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *EMNLP '10*, pages 1162–1172.
- Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *EMNLP '08*, pages 897–906.
- Stefan Evert. 2008. Corpora and collocations. In A. Lüeling and M. Kytö, editors, *Corpus Linguistics. An International Handbook*. Mouton de Gruyter.
- Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. 2013. Multi-step regression learning for compositional distributional semantics. *CoRR*, abs/1301.6939.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING '92*, pages 539–545.
- HIT-SCIR. 2006. Retrieved from: [http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE\\_user\\_op=view\\_page&PAGE\\_id=162](http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=162).
- Alpa Jain and Marco Pennacchiotti. 2010. Open entity extraction from web search query logs. In *COLING '10*, pages 510–518.
- Thorsten Joachims. 1999. Advances in kernel methods. chapter Making large-scale support vector machine learning practical, pages 169–184. MIT Press.
- Nobuhiro Kaji and Masaru Kitsuregawa. 2008. Using hidden markov random fields to combine distributional and pattern-based word clustering. In *COLING '08*, pages 401–408.
- Zornitsa Kozareva, Konstantin Voevodski, and Shang-Hua Teng. 2011. Class label enhancement via related instances. In *EMNLP '11*, pages 118–128.
- Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *ACL '97*, pages 64–71.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *ACL '98*, pages 768–774.
- Diana McCarthy and Roberto Navigli. 2009. The english lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159.
- Shachar Mirkin, Ido Dagan, and Maayan Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *COLING-ACL '06*, pages 579–586.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244.
- Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses*. A Wiley-Interscience publication. Wiley, New York, NY [u.a.].
- Marius Paşca. 2004. Acquisition of categorized named entities for web search. In *CIKM '04*, pages 137–145.
- Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Comput. Linguist.*, 33(2):161–199.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *ACL '06*, pages 113–120.
- Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *COLING '04*.
- Marco Pennacchiotti and Patrick Pantel. 2009. Entity extraction via ensemble semantics. In *EMNLP '09*, pages 238–247.
- Reinhard Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322.
- Shuming Shi, Huibin Zhang, Xiaojie Yuan, and Ji-Rong Wen. 2010. Corpus-based semantic class mining: distributional vs. pattern-based approaches. In *COLING '10*, pages 993–1001.
- Keiji Shinzato and Kentaro Torisawa. 2007. A Simple WWW-based Method for Semantic Word Class Acquisition. In *RANLP 2005*, volume 292 of *Current Issues in Linguistic Theory*, pages 207–216. John Benjamins.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.
- James H. Steiger. 1980. Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87(2):245–251, March.
- Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *ACL '10*, pages 948–957.
- Peter D. Turney. 2006. Similarity of semantic relations. *Comput. Linguist.*, 32(3):379–416, September.
- Fan Zhang, Shuming Shi, Jing Liu, Shuqi Sun, and Chin-Yew Lin. 2011. Nonlinear evidence fusion and propagation for hyponymy relation mining. In *ACL '11*, pages 1159–1168.

# Labeled Alignment for Recognizing Textual Entailment \*

Xiao-Lin Wang Hai Zhao Bao-Liang Lu

Center for Brain-Like Computing and Machine Intelligence  
MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems  
Department of Computer Science and Engineering, Shanghai Jiao Tong University  
800 Dongchuan Road, Shanghai 200240, China  
arthur.xl.wang@gmail.com {zhaohai, blu}@cs.sjtu.edu.cn

## Abstract

Recognizing Textual Entailment (RTE) is to predict whether one text fragment can semantically infer another, which is required across multiple applications of natural language processing. The conventional alignment scheme, which is developed for machine translation, only marks the paraphrases and hyponyms to justify the entailment pairs, while provides less support for the non-entailment ones. This paper proposes a novel alignment scheme, named labeled alignment, to address this problem, which introduces negative links to explicitly mark the contradictory expressions to justify the non-entailment pairs. Thus the alignment-based RTE method employing the proposed scheme, compared with those employing the conventional one, can gain accuracy improvement through actively detecting the signals of non-entailment. The experimental results on the data sets of two shared RTE tasks indicate the implemented system significantly outperforms both the baseline system and all the other submitted systems.

## 1 Introduction

Textual Entailment (TE) is a directional relation between two text fragments. One natural-language premise, noted as  $P$ , entails one natural-language hypothesis, noted as  $H$ , if typically a human read-

ing  $P$  would infer that  $H$  is most likely true (Dagan et al., 2006).

Recognizing Textual Entailment (RTE) is proposed as a generic task that captures the semantic inference need across a wide range of natural language processing applications. For example, a question answering system should identify the texts that entail a hypothesized answer, e.g., given the question “*What does Peugeot manufacture?*”, the text “*Chrétien visited Peugeot’s newly renovated car factory*” entails the hypothesized answer form “*Peugeot manufactures cars*” (Dagan et al., 2006). Similarly, in Machine Translation (MT) evaluation, a correct translation should be semantically equivalent to the gold translation, that is, both translations should entail each other (Padó et al., 2009).

RTE has attracted extensive attention ever since it was proposed. A wide range of methods have been proposed, and quite a few successful approaches treat RTE as an alignment problem. Alignment is originally developed for MT to bridge two languages (Brown et al., 1993). Alignment is to establish links between the semantically equivalent atom expressions in two sentences. (Marsi and Krahrmer, 2005) first advocates pipelined system architectures that contain distinct alignment components. This latter becomes a strategy crucial to the top-performing systems of (Hickl et al., 2006). In addition, human-generated alignment annotations for the second PASCAL<sup>1</sup> RTE challenge is released by Microsoft Research to facilitate related research (Brockett, 2007).

The principle of the existing alignment-based RTE methods is that a sufficiently good alignment between  $P$  and  $H$  means a close lexical and structural correspondence, thus  $P$  probably entails  $H$ . For example, Fig. (1a) shows that the entailment

\* B. L. Lu and X. L. Wang are supported by the National Natural Science Foundation of China (Grant No. 61272248), the National Basic Research Program of China (Grant No.2013CB329401), and the Science and Technology Commission of Shanghai Municipality (Grant No. 13511500200). H. Zhao is supported by the National Natural Science Foundation of China (Grant No. 60903119 and Grant No. 61170114).

<sup>1</sup>PASCAL is the European Commission’s ICT-funded Network of Excellence for Cognitive Systems, Interaction & Robotics.

relation can be correctly predicted through recognizing “*read into*” → “*interpreted*”<sup>2</sup> and “*what he wanted*” → “*in his own way*”.

However, the alignment developed in MT does not solve the non-alignment samples well. It usually links the words in  $H$ , which have no counterparts in  $P$ , to  $NULL$  regardless their impacts on the entailment relation. For example, in Fig. (1b), “*ferry sinking*”, “*cause*” and “*that*” are all linked to  $NULL$ <sup>3</sup>, while only “*ferry sinking*” is the cause for non-entailment. Thus such an alignment is improper for RTE.

This paper extends the normal alignment scheme to meet the challenge of RTE. The proposed scheme, named labeled alignment, introduce another type of links, named negative links, to mark those critical RTE-related linguistic phenomena that cannot be captured by the normal alignment. For example, Fig. (1c) shows that the previous vital expressions “*ferry sinking*” is linked to “*flood*” through a negative link, noted as “*ferry sinking*”  $\nrightarrow$  “*flood*”.

The proposed labeled alignment, which explicitly marks the causes of non-entailment, can facilitate the design of RTE method. This paper proposes an RTE method based on the labeled alignments that actively looks for the signal of negative links in order to correctly recall non-entailment samples.

The main contributions of this paper are as follows,

- A labeled alignment scheme is proposed for RTE;
- An RTE data set annotated with the proposed scheme is released;
- High prediction accuracies are achieved on two RTE data sets.

## 2 Related Work

RTE has attracted extensive attention in the past decade, and a wide range of approaches have been proposed besides the alignment-based methods (Androutsopoulos and Malakasiotis, 2009). The logic-based methods interpret sentences to first-order-logic expressions and then invoke theorem provers (Bos and Markert, 2005). Similarity-based methods employ classifiers to learn from

<sup>2</sup>The notation means that the expression “*read into*” in  $P$  is connected to the expression “*interpreted*” in  $H$ .

<sup>3</sup> $NULL$  means an empty expression.

multiple similarity measures including lexical similarities (Watanabe et al., 2012), edit distance (Rios and Gelbukh, 2012), measurements from MT (Volkh and Neumann, 2011), syntactic tree similarity (Mehdad, 2009) and dependency similarity (Wang and Zhang, 2009). Transform-based methods take entailment as finding a credible transform from the premise to the hypothesis (Kouylekov et al., 2011).

(MacCartney et al., 2008) argues the alignment techniques and tools for MT such as GIZA++ (Och and Ney, 2003) do not readily transfer to RTE. They compare the alignment for RTE with that for MT, and state the following differences:

- The alignment for RTE is monolingual rather than cross-lingual, opening the door to utilizing abundant monolingual resources on semantic relatedness.
- The alignment for RTE is asymmetric, since  $P$  is often much longer than  $H$ .
- One cannot assume approximate semantic equivalence, since  $P$  might be contradictory or independent with  $H$ .
- Little training data is available.

They propose a new alignment tool named MANLI for RTE, but still adopts a alignment scheme similar with the one in MT (Brockett, 2007). This paper, however, revises the alignment scheme to support RTE, especially to address the third difference.

(MacCartney et al., 2006) argues that some critical RTE-related linguistic phenomena such as negations and modalities cannot be captured by alignment. They propose a wide range of features to represent them, and employ a classifier to learn from these specialized features as well as the alignment features to predict the entailment relation. The proposed labeled alignment in this paper, however, can natively process these phenomena, e.g., Fig. (2g) solves negations and (2h) solves modalities.

(Sammons et al., 2010) argues that a single label (whether entailment or not) is insufficient to effectively evaluate the performance of RTE system as well as to guide researchers. They raise a group of detailed entailment phenomena such as simple rewriting rules, lexical relations and passive-active transform, as well as a group of detailed

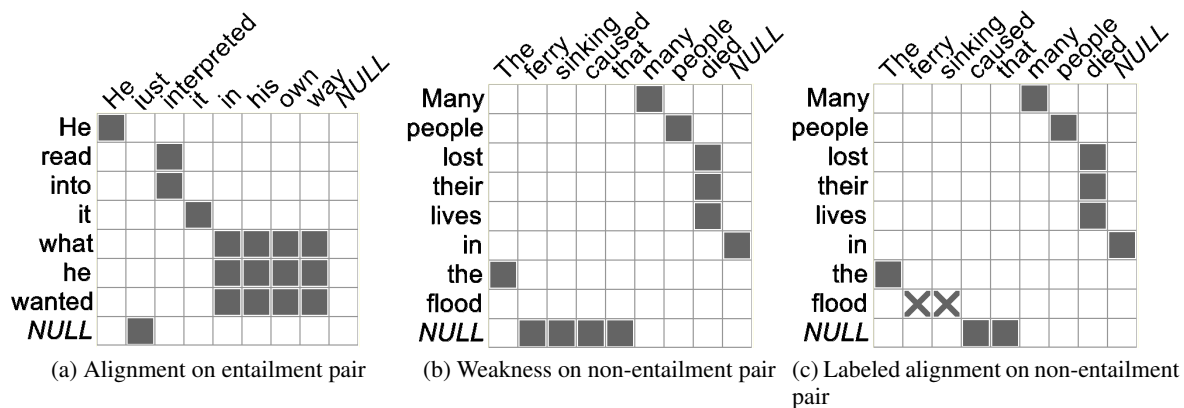


Figure 1: Illustration of Alignment for RTE. Each subfigure presents an RTE sample. The vertical text is the premise, and the horizontal text is the hypothesis. The solid squares represent positive links, and the crosses represent negative links. (a) is of entailment relation, while (b) and (c) are of non-entailment relations.

non-entailment phenomena such as missing arguments, named entities mismatches and missing modifiers. This paper greatly favors their work, and the proposed labeled alignment scheme can annotate most of the non-entailment phenomena mentioned in their paper, which is beneficial to researchers.

### 3 Labeled Alignment

Labeled alignment consists of two types of links, named positive link and negative link, respectively. The positive link is inherited from the normal alignment, while the negative link is newly introduced.

#### 3.1 Positive Link

The positive link is inherited from the normal alignment to handle the variability of natural language expressions, that is, the same meaning can be expressed by different texts. The positive link connects the atom expressions  $e_p$  in  $P$  and  $e_h$  in  $H$ , if  $e_p$  and  $e_h$  are paraphrases or  $e_p$  infers  $e_h$ , noted as  $e_p \rightarrow e_h$ . As the occurrence of this type of links suggests the entailment relation between  $P$  and  $H$ , they are named positive links.

This paper partially follows the alignment scheme in (Brockett, 2007; MacCartney et al., 2008) where the links are token-based but many-to-many is allowed, thus multi-word phrases can be explicitly aligned.

The positive links are mainly applied to the following cases:

- identical words;

- synonyms or near synonyms, e.g., “*bought*”  $\rightarrow$  “*purchased*” in Fig. (2a);
- hyponyms, e.g., “*patent*”  $\rightarrow$  “*technology*” in Fig. (2a);
- same named entities, e.g., “*the Microsoft Corporation*”  $\rightarrow$  “*Microsoft*” in Fig. (2a);
- paraphrases or semantically inferable expressions which cannot be further decomposed into smaller links, e.g., “*read into*”  $\rightarrow$  “*interpreted*” and “*what he wanted*”  $\rightarrow$  “*in his own way*” in Fig. (1a);
- trivial words in  $H$  versus  $NULL$ , e.g.,  $NULL \rightarrow$  “*just*” in Fig. (1a).

#### 3.2 Negative Link

The negative link is introduced to annotate why a RTE sample does not possess an entailment relation. The negative link is noted as  $e_p \not\rightarrow e_n$  where  $e_p$  and  $e_n$  are the expressions in  $P$  and  $H$ , respectively. As the occurrence of this type of links suggests the non-entailment relation, they are named negative links.

The usage of negative links can be divided to three categories – contradictory expressions, unmatched sentence-level modifier and hypothesis novelty.

The contradictory expressions refer to the two expressions from  $P$  and  $H$ , respectively, which should be compared as motivated by the syntactic structures, but actually convey inconsistent semantics. Such phenomena usually lead to the conflic-

tion between  $P$  and  $H$ . The contradictory expressions include, but are not limited to, the following cases:

- antonyms, e.g., “*catalyst*”  $\nrightarrow$  “*deterrent*” in Fig. (2b);
- mismatches between numbers, dates and times, e.g., “*3 millions*”  $\nrightarrow$  “*10,000*” in Fig. (2c);
- different named entities, e.g., “*Mircrosoft*”  $\nrightarrow$  “*Sony*” in Fig. (2d);
- heads of noun phrases, e.g., “*drill*”  $\nrightarrow$   $NULL$  in Fig. (2e);
- vital modifiers of noun phrases, e.g., “*Hispanic*”  $\nrightarrow$   $NULL$  in Fig. (2f);
- contradictory content words<sup>4</sup>, e.g., “*flood*”  $\nrightarrow$  “*ferry sinking*” in Fig. (1c).

The unmatched sentence-level modifier refers to the modifier in either  $P$  or  $H$  which impacts the meaning of the whole sentence but has no counterpart in the other sentence. Such phenomena usually flip the entailment relation. The unmatched sentence-level modifier is marked through connecting it to  $NULL$  through a negative link. The usage includes the following cases:

- negations including simple negation (not), negative quantifiers (no, few), prepositions (without, except), adverbs (never, seldom, nearly), e.g., “*never*”  $\nrightarrow$   $NULL$  in Fig. (2g);
- Virtual modalities, e.g., “*could*”  $\nrightarrow$   $NULL$  in Fig. (2h);
- phrases that suggest the sentence is not stating a happened event, e.g., “*ready to*”  $\nrightarrow$   $NULL$  in Fig. (2i);
- hypothetical conjunctions, e.g., “*if*”  $\nrightarrow$   $NULL$  in Fig. (2j).

The hypothesis novelty refers to the expression in  $H$  that conveys novel information against  $P$ . It is also marked through connecting it to  $NULL$  through a negative link. Such an expression is usually among the following cases:

- numbers, e.g.,  $NULL$   $\nrightarrow$  “*20-30 percent*” in Fig. (2k);
- novel content words, e.g.,  $NULL$   $\nrightarrow$  “*property damage*” in Fig. (2l).

## 4 Alignment-based RTE Methods

In this section, the conventional alignment-based RTE method is introduced first. This method is then augmented to leverage the proposed labeled alignment to improve the prediction accuracy.

### 4.1 RTE Method Based on Normal Alignment

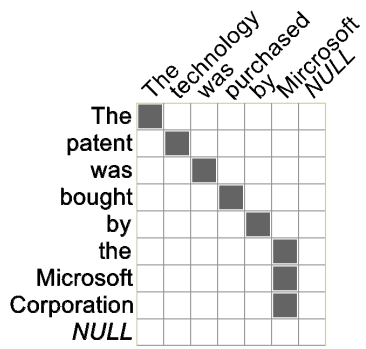
The conventional alignment-based RTE method measures the quality of the alignment between the premise  $P$  and the hypothesis  $H$  to predict their entailment relation (Fig. 3a). An automated aligner is first learned from the annotation of positive links, the normal alignment consists of positive links (see Sec. 3.1). Then this aligner produces an alignment for each input ( $P$ ,  $H$ ). After that, a feature extractor measures the quality of the alignment. Finally a classifier utilizes these measures as features to predict the entailment relation. Commonly used quality measurements for alignment include the confidence score of the aligner and the ratio of linked words in  $P$  (Tab. 1).

### 4.2 RTE Method Based on Labeled Alignment

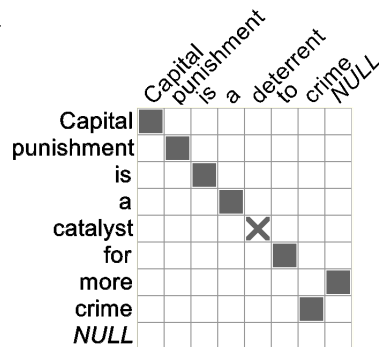
The augmented RTE method based on the labeled alignment not only measures the quality of the alignment, but also detects the signals of negative links to improve the prediction accuracy (Fig. 3b). The augmentation is conducted in two aspects. First, the aligner is trained with both positive and negative links, thus the produced alignment for each input ( $P$ ,  $H$ ) contains both positive and potentially negative links (but two types of links are not distinguished). Second, the feature extractor not only measures the quality of the alignment, but also analyzes the type of each link. A wide range of type-related features can be extracted from each link of the alignment (Tab. 1). These type-related features together with the quality-related features are added into a feature vector for classification.

Notably, besides the above RTE method, a pipeline framework based on the labeled alignment has been tried, but its accuracy turns to be lower than that of the baseline. The

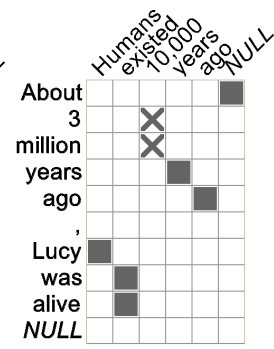
<sup>4</sup>This phenomenon is actually hard to recognize in a practical system. Multiple relevant weak features for classification are employed.



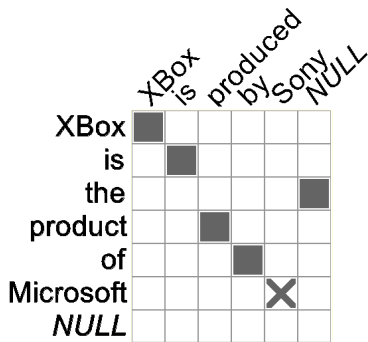
(a) Synonym, hyponym and named entity



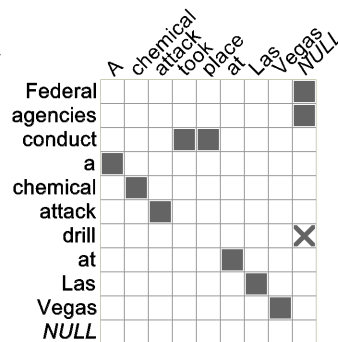
(b) Atonym



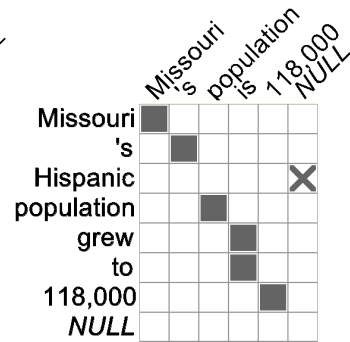
(c) Mismatched numbers



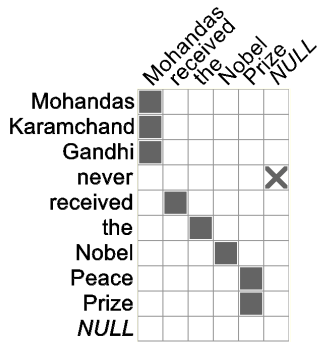
(d) Different named entities



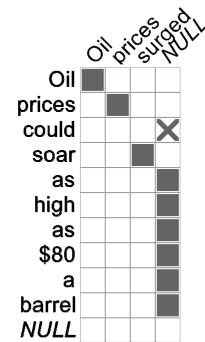
(e) Head of noun phrase



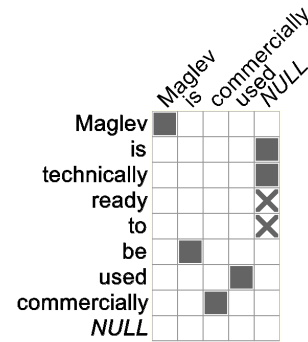
(f) Vital modifier



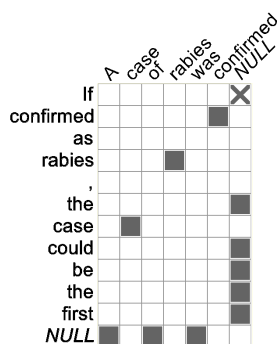
(g) Unmatched negation



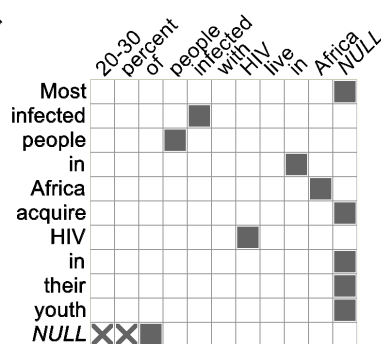
(h) Virtual modality verb



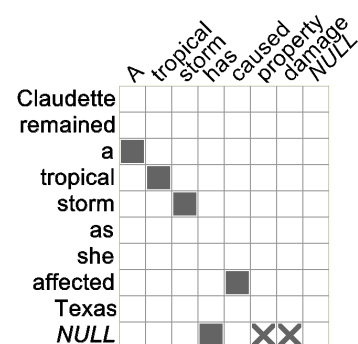
(i) Non-happened event



(j) Hypothetical conjunction



(k) Novel number



(l) Novel content word

Figure 2: Examples of labeled alignment. Each subfigure presents an RTE sample. The vertical text is the premise, and the horizontal text is the hypothesis. The solid squares represent positive links, and the crosses represent negative links. (a) is of entailment relation, and (b)–(l) are of non-entailment relation.

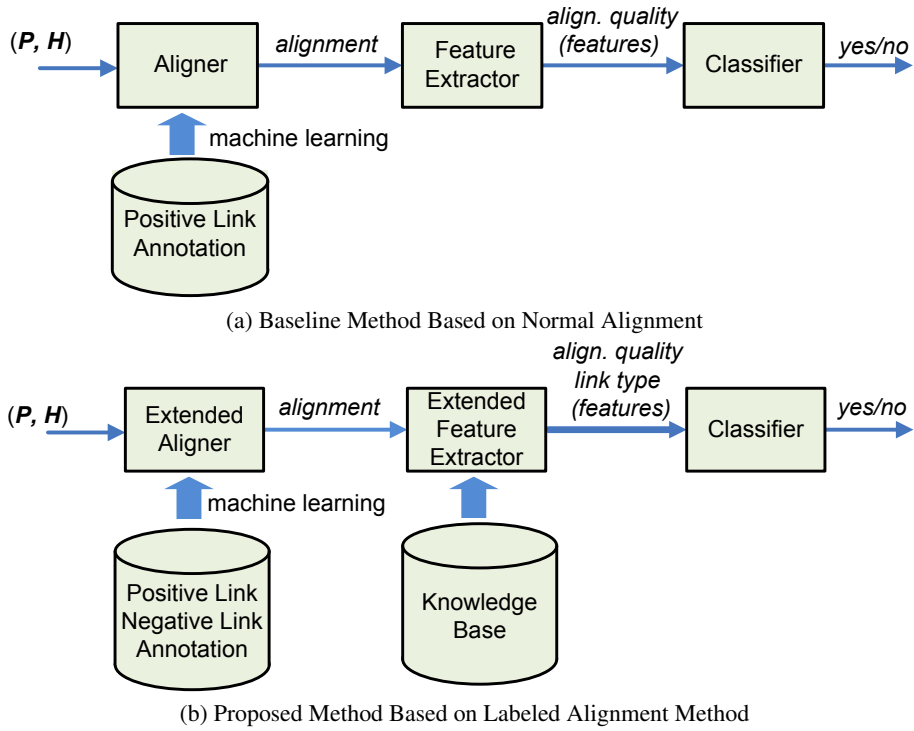


Figure 3: Baseline and Proposed Alignment-based RTE methods

Category	Feature
Align.	Confidence score of the aligner
Quality	Ratio of linked words in $P$
Link Type	Whether $e_P$ and $e_H$ are in an antonym list <sup>a</sup> Whether $e_P$ and $e_H$ are in a synonym list Whether $e_P$ and $e_H$ are unequal numbers Whether $e_P$ and $e_H$ are different named entities Relation of $e_P$ and $e_H$ in an ontology (hyponym, sibling, etc.) Ontology-based similarities of $e_H$ and $e_P$ Count of common characters Length of the common prefixes Length of the common suffix Tuple of the Part-of-Speeches <sup>b</sup> Tuple of the ancestors in an ontology Tuple of whether $e_H$ or $e_P$ is in a list of negative expressions Tuple of whether $e_H$ or $e_P$ is the head of a noun phrase

<sup>a</sup> Suppose the link is from  $e_P$  to  $e_H$  where  $e_P$  and  $e_H$  are the expressions in the premise  $P$  and the hypothesis  $H$ , respectively.

<sup>b</sup> Tuple features are the tuples of the values extracted from  $e_P$  and  $e_H$ , respectively.

Table 1: Features Extracted from Alignments for RTE Classification

	# Train.	# Test.	Ratio Posi.
RITE1	407	407	0.649
RITE2	814	781	0.596

Table 2: Experimental Data Sets

pipeline method first employs a classifier to predict whether each link is positive or negative, and then employs another classifier to predict the entailment relation based on the confidence scores of the first classifier.

## 5 Experiment

The data sets of the NTCIR-9<sup>5</sup> RITE1<sup>6</sup> and NTCIR-10 RITE2 shared tasks (simplified Chinese binary-class track) are taken as the experimental data sets (Shima et al., 2011; Watanabe et al., 2013). This section first describes the annotating process of the labeled alignment, then presents the experimental settings and finally reports the experimental results.

### 5.1 Data Set Annotation

The data sets from the simplified Chinese binary-class tracks of NTCIR-9 RITE1 and NTCIR-10 RITE2 contains 1,595 sentence pairs in all (Tab. 2). Note that all the training and test samples of RITE1 are reused as the training samples of RITE2, while newly collected 781 sentence pairs are taken as the test samples.

The annotating process follows the methodology employed by (Brockett, 2007). The training set of NTCIR-9 RITE1 is used for training annotators, and three Chinese native-speaking undergraduates are actively encouraged to discuss the arising cases, resolve questions and reconcile results with the authors. In annotating the test set of NTCIR-9 RITE1, however, they are first instructed not to discuss the annotations either with the authors or among themselves in order to measure annotator agreement. After that, they reconcile the results on the test set with the authors.

The measure of annotator agreement indicates the alignment annotations are reliably consistent. All three annotators concurred on about 72% of proposed links on the test set, two out of three

<sup>5</sup>NTCIR is the abbreviation of NII Test Collection for IR Systems where NII abbreviates the National Institute of Informatics in Japan.

<sup>6</sup>RITE is the abbreviation of Recognizing Inference in Text.

agreed on about 24% of cases, and three-way disagreements were as rare as about 4%.

### 5.2 Experimental Settings

The supervised learning aligner described in (Chambers et al., 2007) and (MacCartney et al., 2008) is adopted in this paper. This aligner is a structured learning algorithm that employs a linear weighted scoring function to evaluate each candidate alignment. We adapt the original algorithm from two aspects. First, the candidate alignment links are generated from a wide range of NLP analysis results, as follows,

- each segmented word in  $P \rightarrow$  each segmented word in  $H$ ;
- each syntactic node in  $P \rightarrow$  each syntactic node in  $H$ ;
- each NE in  $P \rightarrow$  each NE in  $H$ ;
- each expression  $e_P$  in  $P \rightarrow$  each expression  $e_H$  in  $H$  as long as  $(e_P, e_H)$  appears in a synonym list, a antonym list, or an ontology.

Second, the alignment-learning features contains all the link type features in Tab. 1. These two enhancements, abstractly, convert aligning to a comprehensive NLP process.

The BaseSeg toolkit based on the conditional random field is employed to segment the Chinese texts (Zhao et al., 2006). The Stanford factored parser, which is reported to be more accurate than the PCFG parsers (Klein and Manning, 2002), is employed to analyze the segmented Chinese text. The BaseNER toolkit is employed to recognize named entities (Zhao and Kit, 2008).

We take two Chinese ontologies – CiLin<sup>7</sup> (Mei et al., 1983) and HowNet (Dong and Dong, 2003) – as the knowledge-base for extracting features. Three methods of computing the semantic similarity proposed in (Liu and Li, 2002; Xia, 2007) are employed.

We take the RBF-kernelled SVM as the entailment classifier. The implementation of LibSVM is employed. The parameters are tuned through 5-fold cross-validation on the training set.

The conventional RTE method based on the normal alignment, which is presented in Sec. 4.1, is taken as the baseline method.

<sup>7</sup>This term means a word forest of synonyms in Chinese.



Method	Acc. on RITE1	Acc. on RITE2
Top entries	0.7764 (ICRC_HITSZ <sup>a</sup> Run03 <sup>b</sup> )	0.6850 (MIG Run02) <sup>c</sup>
	0.7617 (FudanNLP Run02)	0.6812 (CYUT Run03)
	0.7568 (ICRC_HITSZ Run02)	0.6658 (WHUTE Run02)
	0.7469 (FudanNLP Run01)	0.6581 (MIG Run01)
	0.7371 (WHUTE Run03)	0.6479 (WHUTE Run01)
normal align. (baseline)	0.7715	0.6991
labeled align. (proposed)	<b>0.8129</b>	<b>0.7465</b>

<sup>a</sup> Team ID;

<sup>b</sup> Run ID. Each team can submit the results of five runs at most.

<sup>c</sup> The top entry is the proposed method, thus not listed.

Table 3: Entailment Prediction Accuracy on NTCIR-9 RITE and NTCIR-10 RITE2 Data Sets

### 5.3 Experimental Results

The experimental results of the prediction accuracy on NTCIR-9 RITE1 and NTCIR-10 RITE2 data sets are presented at Tab. 3. The participants mainly employ committees of classifiers to learn from a wide range of features including multi-level similarities, occurrences of negative words, mismatches of named entities and numbers, syntactic correspondences, and so on (Zhang et al., 2011; Ren et al., 2011). The results show that the proposed RTE method outperforms not only the baseline method, but also the official entries of the shared tasks.

## 6 Conclusion and Future Work

In this paper, a labeled alignment scheme is proposed to address the shortage of the normal alignment scheme for non-entailment RTE samples. To verify the proposed scheme, an augmented alignment-based RTE method that employs the labeled alignment is compared with a conventional one that employs the normal alignment. The data sets of two shared RTE tasks are taken as the experimental data sets and manually annotated with the proposed scheme. Experimental results indicate that the augmented RTE method outperforms not only the baseline method, but also all the submitted systems of the shared tasks. Therefore, the proposed labeled alignment scheme proves to be effective.

The future work of this paper is two-fold. First, during the research, though two Chinese ontology resources – CiLin and HowNet – are employed to detect negative links, it is found that quite a few critical semantic relations are not covered. Therefore we plan to merge and scale existing Chinese ontologies through data mining techniques

such as (Liu and Singh, 2004). Second, the proposed method is actually applicable to multiple languages, though it is only tested on Chinese in this paper. We plan to apply it to other languages such as the Microsoft English RTE corpus in the future.

## References

- Ion Androutsopoulos and Prodromos Malakasiotis. 2009. A survey of paraphrasing and textual entailment methods. <http://arxiv.org/abs/0912.3747>. [accessed 10-Jan-2013].
- Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of HLT-EMNLP*, pages 628–635.
- Chris Brockett. 2007. Aligning the RTE 2006 corpus. *Microsoft Research Technical Report MSR-TR-2007-77*.
- 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.
- Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher D Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.
- Zhen Dong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *Proceedings of International Conference on Natural*

- Language Processing and Knowledge Engineering*, pages 820–824. IEEE.
- Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing textual entailment with LCC’s GROUND-HOG system. In *Proceedings of the Workshop on the Second PASCAL Recognising Textual Entailment Challenge*.
- Dan Klein and Christopher D Manning. 2002. Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, 15(2003):3–10.
- Milen Kouylekov, Alessio Bosca, and Luca Dini. 2011. EDITS 3.0 at RTE-7. *Proceedings of the Seventh PASCAL Recognizing Textual Entailment Challenge*.
- Qun Liu and Su Jian Li. 2002. Computation of semantic similarity for phrases based on HowNet (in Chinese). *Chinese Computational Linguistics*, 7(2):59–76.
- Hugo Liu and Push Singh. 2004. Conceptnet – a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.
- Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of HLT-NAACL*, pages 41–48.
- Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP’08*, pages 802–811.
- Erwin Marsi and Emiel Krahmer. 2005. Classification of semantic relations by humans and machines. In *Proceedings of the ACL workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 1–6.
- Yashar Mehdad. 2009. Automatic cost estimation for tree edit distance using particle swarm optimization. In *Proceedings of ACL-IJCNLP*, pages 289–292.
- Jia Ju Mei, Yi Ming Zhu, and Yun Qi Gao. 1983. *TongYiCi CiLin*. Shanghai Dictionary Publisher.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Sebastian Padó, Michel Galley, Dan Jurafsky, and Chris Manning. 2009. Robust machine translation evaluation with entailment features. In *Proceedings of ACL-IJCNLP*, pages 297–305.
- Han Ren, Chen Lv, and Donghong Ji. 2011. The WHITE system in NTCIR-9 RITE task. In *Proceedings of NTCIR-9 Workshop Meeting, Tokyo, Japan*.
- Miguel Rios and Alexander Gelbukh. 2012. Recognizing textual entailment with a semantic edit distance metric. In *11th Mexican International Conference on Artificial Intelligence*, pages 15–20. IEEE.
- Mark Sammons, VG Vydiswaran, and Dan Roth. 2010. Ask not what textual entailment can do for you... In *Proceedings of ACL’10*, pages 1199–1208.
- Hideki Shima, Hiroshi Kanayama, Cheng-Wei Lee, Chuan-Jie Lin, Teruko Mitamura, Yusuke Miyao, Shuming Shi, and Koichi Takeda. 2011. Overview of NTCIR-9 RITE: Recognizing inference in text. In *Proceedings of NTCIR-9 Workshop Meeting, Tokyo, Japan*.
- Alexander Volokh and Günter Neumann. 2011. Using MT-based metrics for RTE. In *Proceedings of the Seventh PASCAL Recognizing Textual Entailment Challenge*.
- Rui Wang and Yi Zhang. 2009. Recognizing textual relatedness with predicate-argument structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, pages 784–792. ACL.
- Yotaro Watanabe, Junta Mizuno, Eric Nichols, Katsuma Narisawa, Keita Nabeshima, Naoaki Okazaki, and Kentaro Inui. 2012. Leveraging diverse lexical resources for textual entailment recognition. *ACM Transactions on Asian Language Information Processing*, 11(4):18.
- Yotaro Watanabe, Yusuke Miyao, Junta Mizuno, Tomohide Shibata, Hiroshi Kanayama, Cheng-Wei Lee, Chuan-Jie Lin, Shuming Shi, Teruko Mitamura, Noriko Kando, Hideki Shima, and Kohichi Takeda. 2013. Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Workshop. In *Proceedings of NTCIR-10*.
- Tian Xia. 2007. Research on the computation of semantic similarity for Chinese phrases (in Chinese). *Computer Engineering*, 33(6):191–194.
- Yaoyun Zhang, Jun Xu, Chenlong Liu, Xiaolong Wang, Ruifeng Xu, Qingcai Chen, Xuan Wang, Yongshuai Hou, and Buzhou Tang. 2011. ICRC.HITSZ at RITE: Leveraging multiple classifiers voting for textual entailment recognition. In *Proceedings of NTCIR-9 Workshop Meeting, Tokyo, Japan*.
- Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*, pages 106–111.
- Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved Chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165. Sydney: July.

# Context-Based Chinese Word Segmentation using SVM Machine-Learning Algorithm without Dictionary Support

**Chia-ming Lee**

Department of Engineering Science  
and Ocean Engineering,  
National Taiwan University,  
Taipei, Taiwan (R.O.C.)  
trueming@gmail.com

**Chien-Kang Huang**

Department of Engineering Science  
and Ocean Engineering,  
National Taiwan University,  
Taipei, Taiwan (R.O.C.)  
ckhuang@ntu.edu.tw

## Abstract

This paper presents a new machine-learning Chinese word segmentation (CWS) approach, which defines CWS as a break-point classification problem; the break point is the boundary of two subsequent words. Further, this paper exploits a support vector machine (SVM) classifier, which learns the segmentation rules of the Chinese language from a context model of break points in a corpus. Additionally, we have designed an effective feature set for building the context model, and a systematic approach for creating the positive and negative samples used for training the classifier. Unlike the traditional approach, which requires the assistance of large-scale known information sources such as dictionaries or linguistic tagging, the proposed approach selects the most frequent words in the corpus as the learning sources. In this way, CWS is able to execute in any novel corpus without proper assistance sources. According to our experimental results, the proposed approach can achieve a competitive result compared with the Chinese knowledge and information processing (CKIP) system from Academia Sinica.

## 1 Introduction

Chinese sentences contain sequences of characters that are not delimited by white spaces or any other symbol used for word identification, so Chinese word segmentation (CWS) is one of the fundamental issues in Chinese natural language processing studies.

One of the major aspects in existing CWS researches is the resolution of word segment ambiguities. The conventional approach of ambiguity detection is to use two maximum matching methods (MMs), which scan corpora forward

(Forward Maximum Matching, FMM) and backward (Backward Maximum Matching, BMM) based on dictionaries (Kit, Pan, & Chen, 2002). Meanwhile, disambiguation methods can be classified into two different categories: rule-based methods and statistical-based methods. (Ma & Chen, 2003b). Problem disambiguity is often accompanied by the problem resolution of an unknown word or out-of-vocabulary (OOV) extraction (K.-J. Chen & Ma, 2002). Besides the MMs with dictionaries, which are also known as word-based approaches, there are character-based approaches. The word-based approach treats words as the basic unit of a language, and the character-based approach labels each character as the beginning, middle, or end of a word. Character-based approaches are often implemented with a machine-learning classification algorithm for handling disambiguation (Wang, Zong, & Su, 2012). In addition to dictionaries, other linguistic resources such as part-of-speech (POS) or semantic information can be integrated for further improvement (M.-y. Zhang, Lu, & Zou, 2004).

In addition to the disambiguation strategy, many researchers provide the best word sequence identification methods for their CWS. The Hidden Markov model (HMM) (Lin, 2006; M.-y. Zhang et al., 2004), maximum entropy (ME), mutual information (MI) and boundary dependency (Peng & Schuurmans, 2001) are often used. Theoretically, to get the best CWS result is to obtain the optimized word sequence.

As described above, existing CWS research takes either words or characters as the core unit of their methodologies. Instead of identifying word ambiguity, finding word sequence or joining characters into words, we redefine the CWS problem as the identification of “break points” among the “joint points” in Chinese character sequences. In this paper, we define a “joint

point” as a point between adjacent characters, and a “break point” as the boundary of two subsequent words; further, the characters between two break points will consist of words.

The identification of break points among joint points is a binary classification problem. In this study, we use a support vector machine (SVM) machine-learning algorithm with contextual statistical measures to construct the feature vector model of the joint points. Based on our assumption that the Chinese word segmentation rule can be learned from non-linguistic contextual information, all features selected for the joint point model are purely statistical measures without any linguistic tagging information. Moreover, a systematic approach for creating effective positive and negative samples is provided for training the SVM classifier.

Furthermore, in order to meet the need of a CWS approach for a novel corpus, which has no appropriate dictionaries or linguistic tagging, in this study, we select a small set of assistant known source from the experimental corpus as the learning samples, which can be reduced to only 3 words: the most frequent bi-gram, tri-gram, and four-gram words. The experimental results show that by using the joint point model within long contextual information, a small set of learning samples can lead to competitive CWS results compared with the Chinese knowledge and information processing (CKIP) system, which is supported by a large-scale term database that contains approximately 5 million Chinese terms, from Academia Sinica.

## 2 Related Works

Conventionally, ambiguity and OOV are two major problems in the field of CWS research (K.-J. Chen & Ma, 2002). From the methodological perspective, there are rule-based, statistical-based, and machine-learning approaches (Kit et al., 2002; Peng & Schuurmans, 2001; Wang et al., 2012). Moreover, on the basis of the basic language unit used, existing research can be categorized into either word-based or character-based methods (Y. Zhang & Clark, 2007; Zhao, Huang, Li, & Lu, 2010). Most CWS research has resolved problems using labeled corpora while a few have managed CWS using pure text corpora (Dai, Loh, & Khoo, 1999; Jin Kiat Low, 2005). In labeled corpora, the tagging of dictionary matches, parts-of-speech, semantics, and character positions inside a word, are all popular meth-

ods for incorporating known information (Kit et al., 2002).

### 2.1 Ambiguity and the unknown word

There are two types of ambiguities in CWS: overlapping and combinational ambiguities. They can be defined as follows: given a dictionary  $D$  and a string “abc,” if the set of sub-strings  $\{ab, bc\} \subset D$ , “abc” involves an overlapping ambiguity; given a dictionary  $D$  and a string “ab,” if the set of sub-strings  $\{a, b, ab\} \subset D$ , “ab” involves a combinational ambiguity.

Conventional dictionary-based FMM and BMM are straightforward strategies for detecting ambiguities (Kit et al., 2002) and certainly provide an applicable foundation for disambiguation methods. However, dictionaries can never contain all words. Every corpus will have, on average, 3% to 5% OOV words (K.-J. Chen & Ma, 2002); hence, the identification of unknown words has become an important branch of CWS studies (K.-J. Chen & Ma, 2002; Ma & Chen, 2003a). Besides MMs, there are other corpus-based learning approaches to detect ambiguities for CWS (K.-J. Chen & Bai, 1998).

### 2.2 Word-based and character-based approaches

Another way to catalogue CWS is dependent on the basic information unit used; there are both word-based and character-based CWS methods. Word-based approaches treat the word as the basic unit, and POS and other word-based linguistic resources are often integrated into such approaches in order to improve the CWS results. From this point of view dictionary-based approaches can be treated as word-based approaches. Character-based approaches disregard the linguistic information and directly calculate the character-to-character statistical features. One popular way is to label each character as the beginning, middle, or the end of a word, and generate sequence words in sentences on the basis of the position labels of the characters (Goh, 2005; Peng & Schuurmans, 2001; Zhao et al., 2010). There are few character-based CWS approaches that use pure text corpora without additional label information (Dai et al., 1999; Jin Kiat Low, 2005).

### 2.3 Rule-based, statistical-based, and machine-learning methods

From the methods perspective, the earlier CWS used heuristic rules to resolve ambiguities (Ma &

Chen, 2003b) accompanied by the development of unknown word extraction or identification technologies (Ma & Chen, 2003a). Besides rule-based approaches, statistic-based approaches involved the concept of language models trained on large-scale corpora, and many such algorithms have been used and improved over time, such as Maximum Entropy (ME), Mutual Information (MI), and boundary dependency (Jin Kiat Low, 2005; Peng & Schuurmans, 2001). Some statistic-based approaches do not focus on resolving ambiguities, but provide strategies for word sequence identification in sentences. In general, statistical-based approaches tend to provide a generative or discriminative (Wang et al., 2012) probability formula for Chinese words. In contrast, machine-learning approaches pay more attention to the selection of effective features for Chinese word representations. The HMM (Lin, 2006; M.-y. Zhang et al., 2004) and SVM (Li, Huang, Gao, & Fan, 2005) are popular in CWS studies. Currently, a combination of a character-based approach and statistical or machine-learning algorithms is a common strategy for CWS (Goh, 2005; Wang et al., 2012; Zhao et al., 2010).

## 2.4 Contextual information

Dai and Loh have proposed “The Contextual Information Formula” of Chinese bi-gram words (Dai et al., 1999). It is an MI improving formula trained on a large-scale corpus. In this formula, the frequency of a sample bi-gram, the frequencies of its context characters and document frequencies of its context bi-grams are used. They suggest that Chinese words can be defined by a non-linguistic formula that depends on context character measures. Low and Ng conducted a series of studies using context features for their CWS research (Jin Kiat Low, 2005). Further, the concept of contextual information has often been used in information extraction research as well as in existing Chinese term extraction research for entity identification (Gao, 2005; Lee, 2012). In addition, Japanese has no word delimiter like Chinese. Sassano and Neubig et al. have defined Japanese word segmentation (JWS) as a classification task of word boundaries, and also used contextual feature sets in their studies (Neubig, Nakata, & Mori, 2011; Sassano, 2002). Inspired by these ideas of using contextual information, our research aims to extract a contextual information feature vector of “joint points” and uses an SVM algorithm to train a break point classifier.

## 2.5 Complete lexical patterns

Chien has proposed the estimation of complete lexical patterns (Figure 1) (Chien, 1999) in a series of Chinese term extraction papers. There are three important measures used in these lexical patterns, including association, and left and right dependency. These three measures will be integrated into our contextual information feature vector.

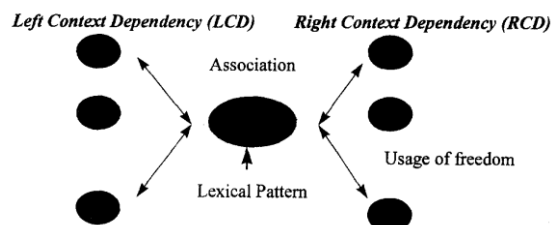


Figure 1. The estimation of complete lexical patterns

- (1)  $Association(AEc) = \frac{f(x)}{(f(y)+f(z)-f(x))}$   
 $x$  is the lexical pattern;  $x = x_1, x_2, \dots, x_n$ ;  $y = x_1, \dots, x_{n-1}$ ,  $z = x_2, \dots, x_n$ ;  $f(x)$  is the frequency of  $x$ ;  $f(y)$  is the frequency of  $y$ ;  $f(z)$  is the frequency of  $z$
- (2)  $Left\ Context\ Dependency(LCD) = \frac{f(\max_{x_l})}{f(x)}$   
 $f(\max_{x_l})$  is the maximum frequency of distinct characters to the left of  $x$
- (3)  $Right\ Context\ Dependency(RCD) = \frac{f(\max_{x_r})}{f(x)}$   
 $f(\max_{x_r})$  is the maximum frequency of distinct characters to the right of  $x$

## 3 Method

In the proposed approach, the CWS was treated as the problem of identifying word break points among joint points in a corpus, and we resolved it by using a SVM machine-learning classifier. The term “joint point” in this paper refers to a point between two adjacent Chinese characters. Our approach is to classify all joint points into either a break point class or non-break point class. The function of break points is similar to that of white spaces in sentences in English.

The SVM is a multi-vector classification algorithm (Boser, Guyon, & Vapnik, 1992). It is also a two-phase algorithm that employs a model-training phase and a model-using (predicting) phase. The major task of the model-training phase is collecting learning samples in different classes and extracting sample feature vectors for training the SVM model. In the model-using phase, the SVM will predict which class an un-

known sample belongs to. Unknown samples need to be formed using the same feature vector as the learning samples. In this paper, we set two classes, the break point class and the non-break point class, and the final predicted break-point outputs are the results of our CWS.

### 3.1 Positive and negative contextual sample generation

In this paper, we propose an efficient method of contextual learning sample generation to build a two class SVM classifier with positive learning samples for the break point class and negative learning samples for the non-break point class. Because break points are the boundaries of words, we first collect the known words in the corpus, and take their boundary points as the positive samples. In contrast, the negative samples are the joint points inside these words. This means that every matching of a word will get two positive learning samples. It will also get one negative (learning sample) for a bi-gram match, two negatives for a tri-gram match and three negatives for a four-gram match.

Take the sentence, "... 我行菩薩道時, ..." (Figure 2), from the experimental corpus as an example, there are nine joint points, p1~p9, in this case. In this sentence, "。" is the period and "，" is the comma in Chinese, and "我行菩薩道時" means 'As I practice the way of Bohdhisattva.' In this case, if 菩薩道 'the way of Bohdhisattva', is a collected known word, then p4 and p7 will be the positive samples and p5 and p6 will be the negative samples; the other joint points will be the unknown samples.

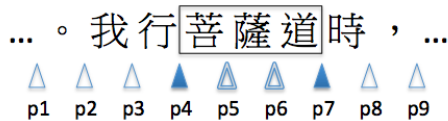


Figure 2. Sample selection example

Joint points, including positive and negative learning samples, and unknown samples, are not characters and therefore do not take up space in a corpus. For making specific samples of the joint points, we always take the same number of characters before and after the joint point to generate contextual samples. Take "p4" in Figure 2 as an example, depending on how long the context information we want to integrate, it can be sampled as a short-distance bi-gram 行菩, which takes one character on both sides of p4, a four-gram 我行菩薩, or a longer-distance n-gram contextual positive sample. In our experiments, a six-gram

contextual sample, catching three characters on both sides of a learning sample, support a better SVM CWS classifier.

The known words, or learning words, for contextual learning sample generation can be collected from dictionaries. However, for the purpose of reducing the preparation loading of CWS for a novel corpus without appropriate dictionaries, in this study, we also set the highest-frequency bi-gram, tri-gram and four-gram words in the corpus for the learning samples generation. Hence, the known words can be collected systematically in this way, and the experiment results suggest that the small size of the known words leads to a competitive result compared with the big numbers of known words, which collected from dictionaries.

The reason for using the most frequent bi-gram, tri-gram and four-gram words, but not uni-gram words is that single-character words do not have a negative case, which would cause an imbalance of positive and negative learning samples. Further, bi-gram words are found to be the majority in Chinese texts, and long words tend to be combinations of short words (梁曉虹, 2005). Further, based on our observation, the highest-frequency bi-gram, tri-gram and four-gram in a Chinese corpus are almost always words and nouns, as well.

### 3.2 Feature vector extraction

The contextual learning sample needs to be modeled as a feature vector for the machine-learning algorithm. There are 10 types of features chosen for the feature vector extraction of the contextual learning sample, including frequency, the number of distinct characters to the left and right, the number of breaking symbols (non-Chinese characters and paragraph marks) to the left and right, association, and the usage freedom to the left and right of characters in the contextual sample. Among these features, association and the usage freedom (also called left and right context dependency) refer to "The Estimation of Complete Lexical Patterns" as proposed by Chien (Chien, 1999). Table 1 shows the complete feature set used in our experiment.

For feature vector extraction, we applied the feature set to every bi-gram plus all uni-gram frequencies within the contextual learning sample. In this way, the long-context feature vector was modeled by measures of short strings, and it could keep particular context information and avoid the probability sparsity to features. Take

the four-gram learning sample, 我行菩薩, generated from p4 in Figure 2 as an example, there are a total of 34 features in its feature vector, including the four uni-gram frequencies of 我, 行, 菩, and 薩, and 30 features from three bi-gram feature sets of 我行, 行菩, and 菩薩. Hence, depending on different extended length of context, there will be 56 features for a six-gram sample and 78 features for an eight-gram sample. Table 2 shows the numbers of features in different lengths of contextual samples.

No.	Features	
1	Frequency	
2	Association (AEc) measure	
3	Number of distinct characters	to the left side
4	Maximum frequency of distinct characters	
5	Number of breaking symbols	
6	Left-context dependency (LCD) measure	
7	Number of distinct characters	to the right side
8	Maximum frequency of distinct characters	
9	Number of breaking symbols	
10	Right-context dependency (RCD) measure	

Table 1. Feature set

Contextual sample length	4	6	8
-uni-grams	4	6	8
-bi-grams	3	5	7
-frequency of each uni-gram	4	6	8
-feature set of bi-grams	30	50	70
Total number of features	34	56	78

Table 2. Number of features in a four-gram feature vector

### 3.3 SVM algorithm and Libsvm package

The Support Vector Machine (SVM) algorithm constructs a hyper-plane in a high-dimensional space for classification and other tasks (Cristianini & Shawe-Taylor, 2000). A good separation is achieved by the hyper-plane farthest from the nearest training data point of any class.

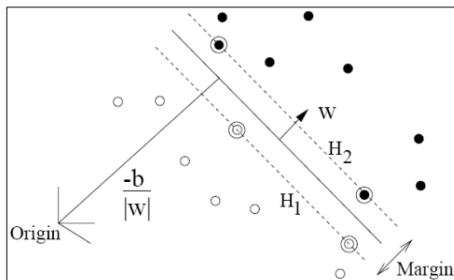


Figure 3. Support Vector Machine (SVM)

In Figure 3,  $W$  is the good separation (the classification hyper-plane) of the two classes—white spots and black spots—and  $H_1$  and  $H_2$  are the support hyper-planes.

$$\mathbf{W}^T \mathbf{X} + b = 0$$

$$H_1: \mathbf{W}^T \mathbf{X} + b = 1$$

$$H_2: \mathbf{W}^T \mathbf{X} + b = -1$$

To maximize the distance between  $H_1$  and  $H_2$  ( $2 / \|\mathbf{w}\|$ ):

$$L(w, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i [y_i (w^T x_i + b) - 1]$$

In our study, Libsvm tools (Chang & Lin, 2011) were used for executing the SVM algorithm. The SVM algorithm includes two phases: model training and model using, called break-point-predicting phases. In the model-training phase, the input data for the Libsvm package is the feature vector data set of all the learning samples, both positives and negatives, and the output data is a classification model file. Meanwhile, all joint points are considered to be unknown samples. The unknown samples should be converted to the feature vector data set in the exactly same way as the learning samples are. In the break-point-predicting phase, the input data is the feature vector data set of the unknown samples and the classification model file output from the earlier phase, and the output data contents of the joint points, which are predicted to be break points. The predicted output data are the CWS results.

## 4 Experiment

### 4.1 Corpus

The collection of Saddharma Puṇḍarīka (Lotus of the True Dharma), which is part of a Chinese text archive from the Middle Ages provided by the Chinese Buddhist Electronic Text Association (CBETA), was selected as the experimental corpus. It consists of 16 sutras labeled T0262 to T0277 of the Taisho Revised Tripitaka. This corpus contains 514,722 Chinese characters without punctuation, and there are a total of 514,721 joint points available for the experiment.

Generally speaking, CWS in ancient Chinese corpora is usually difficult than in modern Chinese collections, as the modern dictionaries are not very suitable for ancient Chinese collections, plus ancient Chinese collections lack punctuations and stop-words. Since the proposed method was designed to solve the CWS without the use of a dictionary, this collection is a good corpus to

demonstrate the powerfulness of the proposed method.

## 4.2 Performance evaluation method

In this study, we selected paragraphs, evaluation texts, from the experimental corpus, and compared the results of the evaluation texts from a subject matter expert's answers and the SVM CWS predicted answers as a means of evaluating the system's performance.

Sātānfēntuólǐjīng, a sutra (T0265) from the collection of Saddharma Puṇḍarīka was chosen as the evaluation text. In Sātānfēntuólǐjīng, there are 1,588 joint points; the ratio size of the evaluation text is 0.3% of the entire corpus. The evaluation text was not included in the training data, and experts provided 616 break points, true answers, and 972 non-break points, false answers for it. Precision, recall, and f-measure were used for evaluating the effectiveness of the CWS results. The evaluation definitions were as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

## 4.3 Experiments

In order to reveal the effectiveness of the training data size, we prepared three training sets; each had different numbers of training data. The first set consisted of learning samples collected from the highest-frequency bi-gram, tri-gram and four-gram, for a total of three Chinese words in the corpus. The second set consisted of learning samples from the top-10 high-frequency words of each bi-gram, tri-gram, and four-gram in the corpus, and the third set consisted of learning samples from 7,309 words, which were collected from the book index of the corpus (大藏經學術用語研究會, 198-?). Table 3 shows the number of learning samples for each training size.

At the end of the experiment, we compared the overall performance with that of CKIP, which is a Chinese word segmentation system supported by 4,892,324 Chinese-word database (Sinica, 2013).

In Table 3, the three highest-frequency words in the first training set are 菩薩 'bodhisattva' (the bi-gram), which had a frequency of 3133; 摩訶薩 'mahasattva' (the tri-gram), which had a frequency of 382; and 文殊師利 'manjushri', a name of the bodhisattvas (the four-gram), which

had a frequency of 514. This is a total of 4,029 matched strings, which contributed 7,658 positive and 5,439 negative samples. Since the matched strings are adjacent in some places, the total number of positives is not exactly twice the summation of the three frequencies. However this does not apply to the negatives samples because there is no commonality of location.

	Highest-frequency words	Top-10 high-frequency words	Dictionary-based group
bi-gram	1	10	2678
tri-gram	1	10	2227
four-gram	1	10	2404
Total words	3	30	7,309
Total positives	7658	35,199	150,441
Total negatives	5439	23,677	105,035

Table 3. Learning sample comparison of three training sets

Besides setting a different size of the training data, we set different context distances, character extensions in context, of samples. The more the context characters are extended, the more is the contextual information involved in the feature vector model. Hence, a two-character extension in context means catching 2 characters on both sides of the joint points to make a 4-gram context learning sample. Table 4 shows the CWS results for the highest-frequency training set, Table 5 shows the results of the top-10 high-frequency training set, and Table 6 shows the results of the dictionary-based training set. Every group was segmented in three different context distances.

Sample length	Four-gram	Six-gram	Eight-gram
Context extension	2 characters	3 characters	4 characters
Precision	51.1%	51.2%	49.5%
Recall	94.5%	94.2%	96.3%
F-measure	66.3%	66.3%	65.3%

Table 4. CWS results of the highest frequency words

In the tables, the dictionary-based results (Table 6) exhibit stable performances; the results growing with context distances. Although the performance of the set of the highest-frequency words is not as good as that of the dictionary-based ones, it is still competitive, and most importantly, it used no assistant sources outside of the corpus.



We believe that this shows the potential of the non-dictionary CWS method proposed in this paper.

Sample length	Four-gram	Six-gram	Eight-gram
Context extension	2 characters	3 characters	4 characters
Precision	56.6%	56.7%	57.0%
Recall	83.4%	82.1%	81.7%
F-measure	67.4%	67.1%	67.2%

Table 5. CWS results of the top 10 high frequency words

Sample length	Four-gram	Six-gram	Eight-gram
Context extension	2 characters	3 characters	4 characters
Precision	57.9%	58.6%	59.1%
Recall	79.5%	80.4%	81.2%
F-measure	67.0%	67.8%	68.4%

Table 6. CWS results of the known words from the index book

#### 4.4 Feature selection analysis

This section analyzes the importance of features used in the SVM classifier. A total of 56 features, in the highest frequency word dataset with 6-gram learning samples, were calculated and sorted by the f-score algorithm proposed by Chen and Lin’s SVM feature-selected research (Y.-W. Chen & Lin, 2006).

Table 7, the top 10 features of the training dataset, shows that the contextual dependency measures around joint points have a significant influence on the SVM classifier.

#### 4.5 Iterative CWS strategy

Because the learning samples can be collected systematically and generated from very few words in the proposed CWS method, we provide an iterative training process to improve the CWS results. In the iterative CWS strategy, we select training samples for the next SVM CWS iterative round from the previous SVM CWS results.

Libsvm provides a probability measure for every joint point in the predicting phase, and in the Libsvm default setting, joint points will be classified in to the break point class when their predicting probability is greater than 50%, which is also the SVM classifier predicting threshold in our experiments.

Based on the probability measure, in the iterative experiment, points whose probability was greater than 90% were taken as positives and

points whose probability was less than 10% were taken as the negatives for the next round. In this way, the size of positives and negatives is imbalanced, so we set a stricter threshold on the side having bigger numbers to make both sides have the same number of learning samples.

Table 8 shows a three-round iterative CWS result using the highest-frequency words training set with the context extension of three characters, the Six-gram learning samples, which led to better performance in the earlier experiment. Based on the performance evaluation over all rounds, the precision in the second round increased by approximately 10%, but other CWS results did not improve as expected.

No.	Features of six-character context sample “ABCDEF”	f-score
1	RCD of “CD”	0.4420
2	LCD of “CD”	0.3304
3	LCD of “DE”	0.3281
4	RCD of “BC”	0.3144
5	Number of distinct characters to the left of “CD”	0.2284
6	Number of distinct characters to the right of “CD”	0.2199
7	AEC of “CD”	0.2108
8	Number of distinct characters to the left of “DE”	0.1598
9	LCD of “BC”	0.1513
10	Number of breaking symbols to the left of “CD”	0.1480

Table 7. Top 10 features of training dataset

	Iteration 1	Iteration 2	Iteration 3
Positives	7658	39520	163849
Negatives	5439	39520	163849
Total learning samples	13097	79040	327698
Precision	51.2%	62.3%	61.6%
Recall	94.2%	66.2%	65.3%
F-measure	66.3%	64.2%	63.4%

Table 8. CWS results of the iterative experiment

#### 4.6 Comparison

Table 9 compares four different CWS results: the highest-frequency words, top-10 high-frequency words, the dictionary group, and the results from CKIP. The best results of each method are shown in this table. CKIP is the segmentation tool by Sinica, which enhances the segmentation using a large-scale term database having approximately 5-million, cross-field Chinese words (Group; Ma & Chen, 2003b). The comparison table shows

that the control group has higher recall, the dictionary-based group has higher precision and the CKIP exhibits a more balanced result.

	Highest frequency words	Top10 high-frequency words	Dictionary	CKIP
Precision	51.2%	57.0%	59.1%	57.4%
Recall	94.2%	81.7%	81.2%	88.3%
F-measure	66.3%	67.2%	68.4%	69.6%

Table 9. Performance comparison

## 5 Conclusion and Discussion

In this paper, we proposed a novel corpus machine-learning CWS approach that identified break points from joint points. The proposed approach is different from existing researches, which tended to create a generating model or formula of Chinese words. In this study, we provided a long-distance context model of joint points and defined the model by non-linguistic contextual features. The experimental results suggested that break points among Chinese texts could be identified on the basis of their non-linguistic contextual features in our chosen corpus.

According to the experimental results, the proposed approach can achieve precision 51.2% and recall 94.2% with only 3 learning words systematically selected from the experiment corpus. It is a very competitive result comparing with the CKIP system, which achieves precision 57.4% and recall 88.3%, and it is supported by an approximately 5-million Chinese-word database. Therefore, this study met the need of carrying out CWS in a novel corpus without appropriate dictionaries.

Further, the proposed approach can systematically select balanced positive and negative learning samples starting from a very small number of learning words. Hence, we chunked long-distance context samples into short-distance strings, uni-grams and bi-grams, for feature vector extraction. Thus, we could collect long-distance context information without dealing with the probability sparsity problem.

Since the CWS rules can be trained from context without linguistic information, the proposed CWS method might also work for Chinese texts from different ages. However, there are some issues and problems that require further investigation.

First, the selection of learning words can affect the final performances. Different learning

words may cause the different results, and this affection needs to be further studied. For instance, if we took learning words by their parts-of-speech instead of frequency, the proposed approach might change its behavior.

Further, the detection of combination words and the overlapping problem needs to be addressed. The Libsvm classifier can assign a predicting probability measure to every joint point. Instead of setting a threshold to filter out break points via these probabilities, these probability measures can be used for identifying the combination words and detecting overlapping problems, as well.

Finally, the effect of iterative process needs to be further studied. Currently, the iterative results can lead to better precision in the second round. However, it performs worse in recall and f-measure. Besides, other iteration parameters need to be decided, such as the number of iteration, the optimal predicting threshold, and the saturation condition for stopping the iterative process properly.

## Acknowledgments

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. 102-2420-H-002-050-MY2 and No. 102-2410-H-002-083-.

## References

- Boser, Bernhard E., Guyon, Isabelle M., & Vapnik, Vladimir N. (1992). *A training algorithm for optimal margin classifiers*. Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, United States.
- Chang, Chih-Chung, & Lin, Chih-Jen. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), 1-27. doi: 10.1145/1961189.1961199
- Chen, Keh-Jiann, & Bai, Ming-Hong. (1998). *Unknown Word Detection for Chinese by a Corpus-based Learning Method*. Paper presented at the Computational Linguistics and Chinese Language Processing.
- Chen, Keh-Jiann, & Ma, Wei-Yun. (2002). *Unknown word extraction for Chinese documents*. Paper presented at the Proceedings of the 19th international conference on Computational linguistics - Volume 1, Taipei, Taiwan.
- Chen, Yi-Wei, & Lin, Chih-Jen. (2006). Combining SVMs with Various Feature Selection Strategies. *Feature Extraction - Studies in Fuzziness and Soft Computing*, 207, 315-324.

- Chien, L. (1999). PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. *Information Processing and Management*, 35(4), 501.
- Cristianini, Nello, & Shawe-Taylor, John. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods*: Cambridge University Press.
- Dai, Yubin, Loh, Teck Ee, & Khoo, Christopher S. G. (1999). *A new statistical formula for Chinese text segmentation incorporating contextual information*. Paper presented at the Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, California, United States.
- Gao, J. (2005). Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4), 531.
- Goh, C. L. (2005). Chinese word segmentation by classification of characters. *International Journal of Computational Linguistics and Chinese Language Processing*, 10(3), 381.
- Group, Chinese Knowledge Information Processing. CKIP Chinese Word Segmentation System, from <http://ckipsvr.iis.sinica.edu.tw/>
- Jin Kiat Low, Hwee Tou Ng, Wenyuan Guo. (2005). *A maximum entropy approach to Chinese word segmentation*. Paper presented at the Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Korea.
- Kit, Chunyu, Pan, Haihua, & Chen, Hongbiao. (2002). *Learning case-based knowledge for disambiguating Chinese word segmentation: a preliminary study*. Paper presented at the Proceedings of the first SIGHAN workshop on Chinese language processing - Volume 18.
- Lee, Chia-ming; Huang, Chien-Kang; Shi, Fayuan; Chen, Kuang-Hua. (2012). *Iterative Chinese Bigram Term Extraction Using Machine-learning Classification Approach*. Paper presented at the The 1st Workshop on Optimization Techniques for Human Language Technology -- Coling 2012, Mumbai, India.
- Li, Hongqiao, Huang, Chang-Ning, Gao, Jianfeng, & Fan, Xiaozhong. (2005). The Use of SVM for Chinese New Word Identification  
Natural Language Processing – IJCNLP 2004. In Keh-Yih Su, Jun'ichi Tsujii, Jong-Hyeok Lee & Oi Kwong (Eds.), (Vol. 3248, pp. 723-732): Springer Berlin / Heidelberg.
- Lin, Qian-Xiang. (2006). *Chinese Word Segmentation using Specialized HMM*. (Master), National Central University, Jhongli.
- Ma, Wei-Yun, & Chen, Keh-Jiann. (2003a). *A bottom-up merging algorithm for Chinese unknown word extraction*. Paper presented at the Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17, Sapporo, Japan.
- Ma, Wei-Yun, & Chen, Keh-Jiann. (2003b). *Introduction to CKIP Chinese word segmentation system for the first international Chinese Word Segmentation Bakeoff*. Paper presented at the Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17, Sapporo, Japan.
- Neubig, Graham, Nakata, Yosuke, & Mori, Shinsuke. (2011). *Pointwise prediction for robust, adaptable Japanese morphological analysis*. Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, Portland, Oregon.
- Peng, Fuchun, & Schuurmans, Dale. (2001). Self-Supervised Chinese Word Segmentation. In Frank Hoffmann, David J Hand, Niall Adams, Douglas Fisher & Gabriela Guimaraes (Eds.), *Advances in Intelligent Data Analysis* (Vol. 2189, pp. 238-247): Springer Berlin Heidelberg.
- Sassano, Manabu. (2002). *An empirical study of active learning with support vector machines for Japanese word segmentation*. Paper presented at the Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania.
- Sinica, Academia. (2013). Academia Sinica Balanced Corpus of Modern Chinese, from <http://db1x.sinica.edu.tw/cgi-bin/kiwi/mkiwi/mkiwi.sh>
- Wang, Kun, Zong, Chengqing, & Su, Keh-Yih. (2012). Integrating Generative and Discriminative Character-Based Models for Chinese Word Segmentation. *ACM Transactions on Asian Language Information Processing* 11(2), 1-41. doi: 10.1145/2184436.2184440
- Zhang, Mao-yuan, Lu, Zheng-ding, & Zou, Chun-yan. (2004). A Chinese word segmentation based on language situation in processing ambiguous words. *Inf. Sci.*, 162(3-4), 275-285. doi: <http://dx.doi.org/10.1016/j.ins.2003.09.010>
- Zhang, Yue, & Clark, Stephen. (2007). *Chinese Segmentation with a Word-Based Perceptron Algorithm*. Paper presented at the Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics.
- Zhao, Hai, Huang, Chang-Ning, Li, Mu, & Lu, Bao-Liang. (2010). A Unified Character-Based Tagging Framework for Chinese Word Segmentation. *ACM Transactions on Asian Language Information Processing*, 9(2), 1-32. doi: 10.1145/1781134.1781135
- 大藏經學術用語研究會. (198-?). *大藏經索引*. 台北: 新文豐.
- 梁曉虹, 徐時儀, 陳五雲. (2005). *佛經音義與漢語詞彙研究*. 北京: 北京商務印書館.

# A Common Case of Jekyll and Hyde: The Synergistic Effect of Using Divided Source Training Data for Feature Augmentation

**Yan Song**

City University of Hong Kong  
83, Tat Chee Ave., Kowloon  
Hong Kong, China  
clksong@gmail.com

**Fei Xia**

University of Washington  
PO Box 354340  
Seattle, WA 98195, USA  
fxia@uw.edu

## Abstract

Feature augmentation is a well-known method for domain adaptation and has been shown to be effective when tested on several NLP tasks (Daume III, 2007). However, a limitation of the method is that it requires labeled data from the target domain and very often such data is unavailable. In this paper, we propose to use training data selection to divide the source domain training data into two parts, pseudo target data (the selected part) and source data (the unselected part), and then apply feature augmentation on the two parts of the training data. This approach has two advantages: first, feature augmentation can be applied even when there is no labeled data from the target domain; second, the approach can take advantage of all the training data including the part that is not selected by training data selection. We evaluate the approach on Chinese word segmentation and part-of-speech tagging and show that it outperforms the baseline where no feature augmentation is applied.

## 1 Introduction

The goal of domain adaptation is to alleviate the degradation of NLP systems when training and test data are from different domains. There have been many approaches to domain adaptation, and two of well-known ones are feature augmentation and training data selection. Feature augmentation makes three copies of each feature in the original feature set (one for the source domain, one for the target domain, and one for the general domain) so that features appeared in the source and the target domains can be differentiated in case they behave differently in the two domains; the method has been shown to be effective for several NLP

tasks (Daume III, 2007). However, a limitation of the method is that it requires labeled data in the target domain, a condition that is hard to meet when creating labeled data in the target domain is expensive and time-consuming.

Training data selection addresses the differences between the source and target domains by choosing a subset of the training data in the source domain that is similar to the data in the target domain. When the amount of source training data is large, this method often provides better performance than using the entire training data (Moore and Lewis, 2010; Axelrod et al., 2011; Plank and van Noord, 2011; Song et al., 2012). However, when the amount of the training data is small, the selected subset is unlikely to outperform the entire training data because the trained model cannot benefit from unselected labeled data.

To address the limitations of both methods, we propose to divide the whole source training data into two subsets via training data selection. We then treat the selected subset as coming from a *pseudo target domain* (i.e., a pseudo domain that is similar to the target domain) and keep the unselected data in the source domain. Now we have labeled data from both domains, we can apply feature augmentation in the usual way; that is, we distinguish features from the source domain and the ones from the pseudo target domain. Notice that the ‘unselected subset’ is also used by the trainer, unlike the standard training data selection method where the unselected part is totally discarded by the trainer. In addition, we propose a coverage-based measure for training data selection. We evaluate our approach on two NLP tasks, Chinese word segmentation (CWS) and part-of-speech (POS) tagging, and show that it outperforms the systems which use the entire training data without training data selection or feature augmentation.

The remainder of this paper is organized as fol-

lows. Section 2 presents previous work on training data selection and feature augmentation. Section 3 describes our approach in details and introduces a coverage-based measure for training data selection. Section 4 reports experimental results on two NLP tasks with discussion on the results.

## 2 Related Work

Two main aspects of our work are dividing training data and applying feature augmentation. In this section, we discuss related work in these aspects.

### 2.1 Training Data Selection

Training data selection is a common approach to domain adaptation. Moore and Lewis (2010) proposed to rank training sentences according to the difference of the cross entropy values of a given sentence, and showed that training data selection improved the performance of statistical machine translation systems. Axelrod et al. (2011) used cross entropy in three ways: the first one directly measured cross entropy for the source side of the text; the second one was similar to (Moore and Lewis, 2010) and ranked the data using cross entropy difference; the third one took into account the bilingual data on both the source and the target side of translations. Both studies showed that the selected subset of training data worked better than the entire training corpus for machine translation. In addition to these studies, there has been other work (e.g., (Eck et al., 2005; Munteanu and Marcu, 2005; Hildebrand et al., 2005; Lu et al., 2007)) that shows training data selection is an effective way to improve MT.

Plank and van Noord (2011) experimented with several training data selection methods to improve the performance of dependency parsing and POS tagging. These methods fell into two categories: probabilistically-motivated and geometrically-motivated. Their experiments demonstrated that the proposed training data selection methods outperformed random selection.

In our previous study (Song et al., 2012), we proposed several entropy-based measures for training data selection, including averaged entropy gain (AEG), cross entropy, difference of entropy, and description length gain (DLG)-based measures. Among them, AEG worked well on CWS and POS tagging and outperformed other measures including difference of cross entropy. In this study, we are using the same data sets as in

that study and we will compare our new coverage-based measure with AEG.

### 2.2 Feature Augmentation

Feature augmentation (Daume III, 2007) is a well-known domain adaptation method in the supervised setting, when labeled data exist for both source and target domains. The idea is to distinguish instances from the source and target domains by making three copies of each original feature: one copy for the source domain, one copy for the target domain and a third copy for the general domain that contains both the source and target domains. Daume evaluated the method on several sequence labeling tasks (e.g., named entity recognition, POS tagging and shallow parsing) and showed that this method outperformed several baselines and previous approaches. The method is easy to implement and does not require modifications to the trainer.

## 3 Our Approach

In order to perform feature augmentation on the whole training data, the very first step is to split the training data into two subsets. Training data selection is an effective way to choose a subset from the whole source domain data that is similar to the target domain. The question is what measures should be used for calculating similarity between a source sentence and the target domain. In this section, we discuss some existing entropy-based measures and propose a novel coverage-based measure. Then we explain how we apply feature augmentation to the two subsets.

### 3.1 Entropy-based Measures

Among the existing similarity measures used by training data selection, many of them focus on the similarity of probability distributions from the training and test data and use entropy-based formulas (Moore and Lewis, 2010; Axelrod et al., 2011; Song et al., 2012). Cross entropy is the most prevailing metric to evaluate the probability distribution similarity between a training sentence and the test data. Eq. 1 shows the formula for cross entropy for a language (marked as CEL, as in the context of evaluating a language model), where  $n$  is the length of sentence  $s$ ,  $p$  is an ngram language model, and  $x_i$  represents the  $i$ -th word in the sen-

tence given the previous words.

$$CEL(s, p) = -\frac{1}{n} \sum_{i=1}^n \log p(x_i) \quad (1)$$

The difference of cross entropy (DCE) for a sentence  $s$  is formulated as

$$DCE(s, p, q) = |CEL(s, p) - CEL(s, q)| \quad (2)$$

where  $p$  and  $q$  are two language models, built from the source domain and the target domain respectively. For training data selection, sentences are sorted by DCE scores and the ones with low scores are considered to be similar to the target domain (Moore and Lewis, 2010; Axelrod et al., 2011).

Another well-performed measure is AEG (Song et al., 2012). Let  $C$  be a corpus and  $s$  be a sentence from the source domain; we define entropy gain (EG) of  $s$  according to  $C$  as in Eq 3, where  $q$  is a probability distribution estimated from  $C$  and  $q_1$  is one estimated from  $C + s$ , a new corpus formed by adding  $s$  to  $C$ . Intuitively, if  $s$  is similar to  $C$ ,  $q_1$  will be very similar to  $q$  and  $EG(s, c)$  will be small.

$$EG(s, C) = |H(C + s, q_1) - H(C, q)| \quad (3)$$

$H(X, p)$  follows the standard definition of entropy in information theory, where  $X$  is a discrete random variable with  $m$  possible outcomes  $\{x_1, \dots, x_m\}$  and  $p$  is a probability distribution of  $X$ . Given a corpus  $C$ , one can collect a set of ngrams (in words or characters) from  $C$  and  $X$  is then derived from the set.

$$H(X, p) = -\sum_{i=1}^m p(x_i) \log p(x_i) \quad (4)$$

Average entropy gain (AEG) is EG normalized by sentence length, shown in Eq 5.

$$AEG(s, C) = \frac{EG(s, C)}{\text{length}(s)} \quad (5)$$

### 3.2 Coverage-based Data Selection

We propose a coverage-based measure, which differs from the entropy-based measures in two aspects. First, this measure uses ngram coverage, not probability similarity, as the criterion for selecting training data. The rationale is that we would like the selected data to have a good coverage of the test data, because in many NLP tasks, especially in CWS and POS tagging, out-of-vocabulary (OOV)

is a main problem affecting system performance and the problem is more severe when the training and test data come from different domains. Second, existing training data selection methods (such as the ones listed in Section 3.1) select the current sentence without considering the effect of adding that to the previously selected sentences. Our method tackles this problem by considering the overall effect of the selected subset. As checking all the subsets is computationally expensive, we use a greedy search to find the best training sentence based on the current selected subset.

The coverage-based data selection is presented in Algorithm 1. Here,  $L$ ,  $T$ , and  $p$  refer to the original training data, test data and the proportion (in percentage) of training data to be selected.  $L_s$  and  $L_u$  are the output, which refer to the selected and unselected subsets of the training data respectively. By conducting such selection method, training data is divided into two parts.

---

**Algorithm 1** Coverage-based data selection.

---

**Input:**  $L, T, p$

**Output:**  $L_s, L_u$

```

1:  $L_s = \phi, L_u = L$ 
2: while  $Sizeof(L_s) < Sizeof(L) * p$  do
3:   for each sentence  $s_i$  in  $L_u$  do
4:     compute  $cov(L_s \cup \{s_i\}, T)$ 
5:   end for
6:    $id = argmax_i cov(L_s \cup \{s_i\}, T)$ 
7:    $L_s = L_s \cup \{s_{id}\}, L_u = L_u - \{s_{id}\}$ 
8: end while
9: return  $L_s, L_u$ 

```

---

In Algorithm 1, coverage function  $cov(C, T)$  represents the coverage of ngrams in a test data  $T$  given a corpus  $C$ , as shown in Eq. 6. Here,  $ng$  is an ngram<sup>1</sup> and  $NgramSet(T)$  refers to the set of ngram types in  $T$ , and the denominator  $|NgramSet(T)|$  is the size of the set.<sup>2</sup>

$$cov(C, T) = \frac{\sum_{ng \in NgramSet(T)} count(ng, C)}{|NgramSet(T)|} \quad (6)$$

To handle the problem of data sparsity, we use the following back-off counting method to find

<sup>1</sup>Where the units for composing an ngram are different with respect to different tasks, i.e., they are characters in word segmentation and words in POS tagging.

<sup>2</sup>We also investigated using ngram tokens for coverage computation; we will include the comparison of ngram types and ngram tokens in the final version of this paper.

partial covered low order ngrams inside the high order ngram. The idea of such ngram counting is similar to back-off methods in language modeling. Given an ngram  $t_{i-n+1} \dots t_{i-1} t_i$  in  $T$ , we calculate the  $count()$  function as in Eq. 7.  $\alpha$  is used to determine the value of the ‘‘partial credit’’ given to a substring of the ngram appearing in  $C$ . The value of  $\alpha$  is set to 0.5 empirically.<sup>3</sup>

$$count(t_{i-n+1} \dots t_{i-1} t_i, C) = \begin{cases} 1, & \text{if } t_{i-n+1} \dots t_{i-1} t_i \text{ appears in } C \\ \alpha \cdot count(t_{i-n+2} \dots t_{i-1} t_i), & \text{otherwise} \end{cases} \quad (7)$$

In Eq. 7,  $t_i$  is a token in the ngram, i.e., a character in the CWS task and a word in the POS tagging task. If a high order ngram is not found in  $C$ , the  $count()$  function is called recursively until a shorter ngram inside the original ngram is found. The value of the  $count()$  function is zero only if the token  $t_i$  itself is an OOV. For the experiments in this paper, we use trigram to count the ngram coverage.

### 3.3 Feature Augmentation

As we mentioned before, a limitation of feature augmentation (Daume III, 2007) is that it requires labeled data from the target domain, and very often such data is not available. To overcome this limitation, we use training data section on the source domain data, treat the selected part of data as from a *pseudo target domain*, and leave the unselected part in the source domain. Then a feature augmentation is performed on such two ‘‘new’’ domains; that is, it makes three copies of each original feature:  $f_s$  for the source domain,  $f_t$  for the target domain, and  $f_g$  for the general domain. Following Daume (Daume III, 2007), the general domain is simply the union of the source and the target domains. In this case, the target domain refers to our pseudo target domain; the features associated to the pseudo target domain and the test data are augmented as in Eq. 8, and the features associated to the unselected source domain data are shown in Eq. 9.

$$f \rightarrow \langle f_g, 0, f_t \rangle \quad (8)$$

<sup>3</sup>We tried different value of  $\alpha$  in ranging from 0 to 1, where  $\alpha = 0$  means there is no back-off. The results indicate that when  $\alpha = 0$ , selection performance is much worse than the case  $\alpha > 0$ , while when  $\alpha > 0$ , selection performance varies so little by using different values of  $\alpha$ .

$$f \rightarrow \langle f_g, f_s, 0 \rangle \quad (9)$$

Another potential issue with feature augmentation is that making several copies of all the features could worsen the problem of data sparsity. It is worth exploring whether duplicating only certain features would produce better performance than duplicating all the features. To test out the idea, we ran another set of experiments where only unlexicalized features (e.g., word type, POS tags of previous words) are duplicated. The experimental results in Section 4 confirmed our intuition and showed that augmenting only unlexicalized features works better.

## 4 Experiments

In this study, we ran several sets of experiments. We compared our training data selection with other methods, and then evaluated our revised feature augmentation method on the CWS and POS tagging tasks.

### 4.1 Data

The Chinese Penn Treebank (CTB) version 7.0<sup>4</sup> (Xia et al., 2000) is used in our experiments. It contains about 1.2 million words from five genres: Broadcast Conversation (BC), Broadcast News (BN), Magazine (MZ), Newswire (NW), and Weblog (WB). The details of the five genres of CTB 7.0 are shown in Table 1.

We divide the data in each genre into ten folds based on character counts, and use the first eight folds for training, the next fold for development, and the last fold for testing. In order to make the size of the training data for each genre to be the same, we set the training size to be the size of the training folds in the BC genre (the smallest genre in the CTB 7.0). We do the same for the development data. For testing, we use the whole test fold for each genre. The sizes of the data sets used in the experiments are shown in Table 2.<sup>5</sup>

Without loss of generality, we use BC and NW as the test genres; for each test genre, we use the union of training folds from other four genres as the training data.

<sup>4</sup>Linguistic Data Consortium No. LDC2010T07

<sup>5</sup>Although we are not using the development fold for the experiments in this study, we still split the data into training, development, and test folds to facilitate comparison with other studies that use the same data split.

Genre	# of chars	# of words	# of files	Sources
Broadcast Conversation (BC)	275,289	184,161	86	China Central TV, CNN, MSNBC, Phoenix TV, etc.
Broadcast News (BN)	482,667	287,442	1,146	China Broadcasting System, China Central TV, China National Radio, Voice of America, etc.
Magazine (MZ)	402,979	256,305	137	Sinaroma
Newswire (NW)	442,993	260,164	790	Xinhua News, Guangming Daily, People’s Daily, etc.
Weblog (WB)	342,116	208,257	214	Newsgroups, Weblogs
Total	1,946,044	1,196,329	2,373	

Table 1: Statistics of the CTB 7.0.

	BC	BN	MZ	NW	WB
Training	211,795	211,826	211,834	211,853	211,796
Development	30,678	30,760	30,708	30,726	30,746
Test	32,816	48,317	37,531	44,543	33,623

Table 2: Statistics of training, development, and test portions of each genre in CTB 7.0. The numbers are character counts.

## 4.2 Training Data Selection

To demonstrate our coverage-based training data selection method, we first compare its performance on POS tagging with other two methods, AEG (Song et al., 2012) and random selection.<sup>6</sup> The selected proportion of training data range from 10% to 90%, based on character counts. Here, we use Stanford POS Tagger (Toutanova et al., 2003). The results on BC and NW are shown in Table 3 and 4, with comparison to random selection methods.<sup>7</sup>

Our coverage-based training data selection method outperforms random selection on both BC and NW. It also outperforms AEG when a low percentage of data is selected, while its performance is comparable or slightly lower than AEG when a higher percentage of data is selected. To understand this behavior, we compare some statistics of the data sets, as in Table 5.

Since OOV rate is important for CWS and POS tagging, we want to compare our coverage-based method and AEG for this factor, and the results are presented in Table 6.

The table shows that when a small percentage

<sup>6</sup>Song et al. (2012) showed that AEG works better than cross entropy, as well as difference of cross entropy, on CWS and POS tagging. Therefore we only compare our method with AEG in this paper.

<sup>7</sup>For each percentage, the result of random selection are the average of three runs of random selection.

(e.g., 10%, 20%) of source-domain data is selected, the OOV rate of the test data is much lower when Cov is used. In contrast, when a large percentage (e.g., 80% and 90%) of training data is selected, the OOV rates are similar between Cov and AEG. This could be the reason why Cov outperforms AEG when a small percentage of training data is selected, but not so when more training data is selected.

For the rest of the experiments, we will use Cov for training data selection and test whether our revised feature augmentation approach provides some improvement for CWS and POS tagging.

## 4.3 Chinese Word Segmentation

To evaluate feature augmentation on CWS, we use a conditional random fields (CRF) word segmenter as described in (Song and Xia, 2012). A nice property of the segmenter is that it incorporates unsupervised learning to identify possible new words in the test data in order to enhance the segmenter’s performance on OOVs. To be more specific, the segmenter uses description length gain (DLG) (Kit and Wilks, 1999) for lexical acquisition as that was performed in (Kit, 2000; Kit, 2005). Then the decision of the unsupervised word segmentation is represented as features  $T_0^i$ , which indicates the tag of the current character  $C_0$  when it belongs to a word whose length  $i$  ranges from 1 to 5 charac-



Percentage	Cov	AEG	RDM
10%	<b>90.08</b>	89.61	88.60
20%	<b>91.13</b>	91.01	89.74
30%	<b>91.40</b>	<b>91.40</b>	90.59
40%	<b>91.70</b>	91.67	91.25
50%	91.89	<b>91.94</b>	91.37
60%	92.24	<b>92.31</b>	91.84
70%	92.40	<b>92.53</b>	91.84
80%	<b>92.43</b>	92.41	92.11
90%	<b>92.48</b>	92.45	92.22
100%	92.30	92.30	92.30

Table 3: Performance of Stanford POS tagger when tested on BC and trained on the other four genres. The largest number in each row is in bold. Cov, AEG and RDM refer to our coverage-based method, Average entropy gain and random selection.

Percentage	Cov	AEG	RDM
10%	<b>89.97</b>	87.73	87.53
20%	<b>91.15</b>	89.64	89.23
30%	<b>91.73</b>	90.74	90.31
40%	<b>91.91</b>	91.41	91.32
50%	<b>92.21</b>	91.86	91.38
60%	<b>92.18</b>	92.03	91.63
70%	<b>92.32</b>	92.19	91.90
80%	92.41	<b>92.45</b>	92.28
90%	<b>92.51</b>	92.48	92.33
100%	92.56	92.56	92.56

Table 4: Performance of Stanford POS tagger when tested on NW and trained on the other four genres. The largest number in each row is in bold. Cov, AEG and RDM refer to our coverage-based method, Average entropy gain and random selection.

Test genre	BC	NW
Tokens in training	536,356	533,594
Tokens in test	22,088	25,916
OOV tokens	1,034	1,986
OOV rate	4.68%	7.66%

Table 5: Statistics (in words) of the entire training and test data for BC and NW.

%	BC		NW	
	Cov	AEG	Cov	AEG
10%	9.04%	11.53%	14.27%	19.61%
20%	5.22%	8.21%	10.19%	14.59%
80%	4.76%	5.19%	7.66%	8.18%
90%	4.68%	4.85%	7.66%	7.94%

Table 6: The OOV rate (in words) when a different percentage (10%, 20%, 80% and 90%) of training data is selected by coverage-based method (Cov) and AEG against test data on BC and NW.

ters. These features are added to the standard feature set for supervised learning. The new feature set is in Table 7, where the subscript -1, 0, and +1 refer to the previous, current and next character, respectively.

Description	Features
Char Unigrams	$C_{-1}, C_0, C_{+1}$
Char Bigrams	$C_{-1}C_0, C_0C_{+1}, C_{-1}C_{+1}$
DLG Features	$T_0^1, T_0^2, T_0^3, T_0^4, T_0^5$

Table 7: Feature template of our CRF segmenter.

For feature augmentation, we compare two settings: one duplicates all the features and the other duplicates only the unlexicalized features. The results when tested on BC are in Table 8. It shows that augmenting unlexicalized features provides better performance than augmenting all features. For the rest of experiments, feature augmentation will duplicate only the unlexicalized features.

Table 9 shows the performance of using feature augmentation on CWS when tested on NW. Table 8 and 9 both show that our approach on divided training data improves system performance significantly (e.g., over 0.6% when tested on BC) without using any external resources. For Tables 9 and 11, we use a ten-partition two-tailed paired Student t-test for significance test.

#### 4.4 POS Tagging

To evaluate feature augmentation on POS tagging, we used an in-house CRF tagger.<sup>8</sup> Table 10 shows the feature set used by the tagger, where subscript -1, 0, and +1 refer to the previous, current and

<sup>8</sup>The reason that we use our in-house CRF POS tagger, instead of the Stanford POS tagger, is that we have not found an easy way to extend Stanford POS tagger to support feature augmentation.

% of data selected	Unlex. Feat. Aug.			All Feat. Aug.		
	F	P	R	F	P	R
Baseline	94.10	93.87	94.34	94.10	93.87	94.34
10%	<b>94.70</b>	94.30	95.09	93.71	93.43	94.00
20%	<b>94.72</b>	94.35	95.09	94.06	93.98	94.14
30%	<b>94.62</b>	94.23	95.01	<b>94.19</b>	94.07	94.31
40%	<b>94.51</b>	94.07	94.96	<b>94.11</b>	93.96	94.26
50%	<b>94.51</b>	94.08	94.94	93.96	93.80	94.12
60%	94.10	93.77	94.43	93.96	93.86	94.06
70%	<b>94.16</b>	93.86	94.46	94.06	93.99	94.12
80%	94.08	93.80	94.37	93.88	93.81	93.95
90%	94.08	93.84	94.32	93.90	93.60	94.20

Table 8: Performance of feature augmentation on CWS, with unlexicalized and all features augmented. The pseudo target data is selected by coverage-based method. The segmenter is tested on *BC*, and trained on the other four genres in CTB 7.0. F-score (F), Precision (P) and Recall (R) are presented. F-scores higher than the baseline are in bold.

%	F	P	R
Baseline	93.70	93.90	93.50
10%	<b>93.82</b>	93.97	93.66
20%	<b>93.90*</b>	94.07	93.73
30%	<b>93.90*</b>	94.05	93.76
40%	<b>93.92**</b>	94.06	93.78
50%	<b>93.89*</b>	94.07	93.71
60%	<b>93.91**</b>	94.09	93.72
70%	<b>93.91**</b>	94.07	93.76
80%	<b>93.89*</b>	94.06	93.72
90%	<b>93.84</b>	94.03	93.64

Table 9: Performance of feature augmentation on CWS, with unlexicalized features augmented. The pseudo target data is selected by coverage-based method. The segmenter is tested on *NW*, and trained on the other four genres in CTB 7.0. F-score (F), Precision (P) and Recall (R) are presented. F-scores higher than the baseline are in bold. Symbols \* and \*\* indicate significance at  $p=0.05$  and  $p=0.01$  against the baseline, respectively.

next word, respectively. This feature set is similar to the one used in the Stanford POS tagger, but our tagger does not include some hard coded treatment and rules (e.g., bidirectional transition rules) used by the Stanford tagger. As a result, the performance of our tagger is slightly lower than the Stanford tagger. For instance, when tested on *BC* and trained on the other four genres, the tagging accuracy of our tagger is 91.95%, compared to 92.30% by the Stanford tagger (see the last row

Description	Features
Word Unigrams	$W_{-1}, W_0, W_{+1}$
Word Bigrams	$W_{-1}W_0, W_0W_{+1}, W_{-1}W_{+1}$
Word Prefix	$P_0$
Word Suffix	$S_0$
Word Prefix Type	$TP_0$
Word Suffix Type	$TS_0$

Table 10: Feature template of our CRF POS tagger.

in Table 11 and Table 3).

Table 11 shows the results of POS tagging with feature augmentation. The test genre is *BC* or *NW*, and the training data come from the other four genres. The first row lists the percentage of training data chosen by our coverage-based training data selection. The baseline shows the performance of our CRF tagger when the whole training set is used without training data selection and feature augmentation. In the table, the higher-than-baseline tagging accuracy in each test are marked in bold-face. Similar to CWS, training data selection followed by feature augmentation improves the performance of the POS tagger.

#### 4.5 Discussion

In all, there are several observations from Tables 8 and 9 for CWS, and Table 11 for POS tagging. First, there is a small, but statistically significant, improvement when we treat selected and unselected data as two domains and apply fea-

Percentage	BC	NW
10%	<b>92.21</b>	92.21
20%	<b>92.31*</b>	<b>92.41</b>
30%	<b>92.40**</b>	<b>92.52*</b>
40%	<b>92.39**</b>	<b>92.48*</b>
50%	<b>92.44**</b>	<b>92.44</b>
60%	<b>92.43**</b>	<b>92.42</b>
70%	<b>92.45**</b>	<b>92.38</b>
80%	<b>92.40**</b>	92.33
90%	<b>92.31*</b>	92.31
baseline	91.95	92.36

Table 11: Performance of our POS tagger with feature augmentation when tested on BC and NW. Numbers presented in the table are tagging accuracy, and the ones higher than the baseline are in bold. Symbols \* and \*\* indicate significance at  $p=0.05$  and  $p=0.01$  against the baseline, respectively.

ture augmentation (e.g., 91.95% vs. 92.45% on BC in Table 11). Second, duplicating only a subset of features outperforms duplicating all the features, as the large number of features for the latter strategy could aggravate the data sparsity problem. Augmenting some features (e.g., lexicalized) could actually hurt the performance. Third, with regard to the percentage of training data selected for the pseudo target domain, system performance improves when the percentage of selected data increases from 10% up to a certain point (70% for testing on BC and 30% for testing on NW on POS tagging), and afterwards it starts to degrade because newly added pseudo target domain data is no longer quite similar to the target domain. The optimal size of the selected subset may depend on how similar the training data is to the test data. Fourth, when comparing CWS and POS tagging, we can find the same trend in feature augmentation across different tasks. That is, when feature augmentation on CWS has higher improvement, usually it also brings higher improvement on POS tagging when comparing across different test data (e.g., the improvement on BC is higher than NW for CWS, and the same is true for POS tagging).

## 5 Conclusion

This study has made two contributions to domain adaptation. First, we proposed an approach that combines training data selection and feature augmentation. It tackles the limitations of both feature

augmentation and training data selection methods as it does not require labeled data from the target domain while it takes advantage of the entire training data. Consequently, it significantly improves system performance over the baseline. We also demonstrate that augmenting some features works better than augmenting all the features because the latter setting triples the number of features which could lead to severe data sparsity problem. Our experimental attempts confirmed the fact that augmenting less-sparse features (unlexicalized one, e.g., prefix and suffix, character type) led to better performance than all features. Second, we proposed a new measure for training data selection, which selects training sentences to maximize the coverage of ngrams on the test data. It showed a better performance than other measures especially when a small subset of training data is selected. The approaches has been evaluated on two NLP tasks, namely, Chinese word segmentation and part-of-speech tagging. Both tasks confirmed the effectiveness of our approaches and yield better performance than the baseline settings.

For future work, we would like to apply automatic feature selection to determine what kind of features should be duplicated to boost the benefits of feature augmentation. We would also like to evaluate our approach on other NLP tasks, and test its performance with other machine learning algorithms.

## References

- Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP2011*, pages 355–362.
- Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 256–263.
- Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of MT Summit X*, pages 227–234.
- Almut Silja Hildebrand, Matthias Eck, and Stephan Vogel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT 2005*, pages 133–142.
- Chunyu Kit and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In *Proceedings of CoNLL-99*, pages 1–6.

- Chunyu Kit. 2000. *Unsupervised Lexical Learning as Inductive Inference*. Ph.D. thesis, University of Sheffield.
- Chunyu Kit. 2005. Unsupervised lexical learning as inductive inference via compression. In J. W. Minett and W. S.Y. Wang, editors, *Language Acquisition, Change and Emergence*, pages 251–296.
- Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of EMNLP-CoNLL2007*, pages 343–350, Prague, Czech Republic, June. Association for Computational Linguistics.
- R.C. Moore and W. Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of ACL2010, Short Papers*, pages 220–224.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.
- B. Plank and G. van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 1566–1576.
- Yan Song and Fei Xia. 2012. Using a goodness measurement for domain adaptation: A case study on chinese word segmentation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3853–3860, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1580.
- Yan Song, Prescott Klassen, Fei Xia, and Chunyu Kit. 2012. Entropy-based training data selection for domain adaptation. In *Proceedings of COLING 2012*, pages 1191–1200, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, pages 173–180.
- Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fudong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*.

# Detecting Polysemy in Hard and Soft Cluster Analyses of German Preposition Vector Spaces

Sylvia Springorum, Sabine Schulte im Walde and Jason Utt

Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart

Pfaffenwaldring 5b, 70569 Stuttgart, Germany

{riestesa, schulte, uttjn}@ims.uni-stuttgart.de

## Abstract

This paper presents a methodology to identify polysemous German prepositions by exploring their vector spatial properties. We apply two cluster evaluation metrics (the *Silhouette Value* (Kaufman and Rousseeuw, 1990) and a fuzzy version of the *V-Measure* (Rosenberg and Hirschberg, 2007)) as well as various correlations, to exploit hard vs. soft cluster analyses based on Self-Organising Maps. Our main hypothesis is that polysemous prepositions are outliers, and thus represent either (i) singletons or (ii) marginals of the clusters within a cluster analysis. Our analyses demonstrate that (a) in a subset of the clusterings, singletons have a tendency to contain polysemous prepositions; and (b) misclassification and cluster membership rate exhibit a moderate correlation with ambiguity rate.

## 1 Introduction

Vector space models have become a steadily increasing, integral part of data-intensive lexical semantics over the past 20 years (cf. Turney and Pantel (2010) and Erk (2012) for two recent surveys). They have been exploited in psycholinguistic (Lund and Burgess, 1996) and computational linguistic research (Schütze, 1998), to explore distributional properties of target objects and the notion of “similarity” within a geometric setting.

While individual vector space approaches have been concerned with sense discrimination, it is still largely unknown how to identify polysemous objects within a vector space model, and which geometric properties characterise the polysemous objects. For example, Schütze (1998) performed sense discrimination of ambiguous word tokens, based on their second-order co-occurrence

distributions; Erk (2009) presented two variants of defining regions of word meaning in vector spaces; Erk and Padó (2010) defined a model where polysemous words activated several word vectors; Boleda et al. (2012b) compared two models of representing regular polysemy, one with multiple class assignments for multiple senses, and one incorporating classes with polysemy properties. Our work is different from all these approaches, since we aim to investigate prototypical spatial properties of polysemous objects.

More specifically, this paper is part of a larger framework that systematically explores the vector spatial properties of German prepositions, a notoriously polysemous closed word class. Relying on Self-Organising Maps (SOMs, cf. Kohonen (2001)) and preposition-dependent nouns as vector-space features, we present a methodology to identify the degree of polysemy of the prepositions. For this task, the methodology applies two cluster evaluation metrics, the *Silhouette Value* (Kaufman and Rousseeuw, 1990) and the *V-Measure* (Rosenberg and Hirschberg, 2007), to hard vs. soft cluster analyses based on the Self-Organising Maps. Since we start out with a hard clustering, a sub-task is concerned with transferring the SOM hard clusters to soft clusters. Similarly, the original V-Measure applies to hard clusters only, so a second sub-task is concerned with defining a Fuzzy V-Measure that applies to soft clusters. Our main hypothesis is that polysemous prepositions are outliers, and thus represent either (i) singletons or (ii) marginals of the clusters within a cluster analysis.

The paper is organised as follows. After introducing our preposition data in Section 2, Section 3 describes the preposition vector-space features, and the hard and soft clusterings. Section 4 is devoted to the evaluations, and Section 5 relies on the cluster analyses and the evaluations, to detect and discriminate polysemous prepositions.

## 2 Preposition Data

Although prepositions contribute a considerable portion to the meaning of texts, comparably little effort in computational semantics has gone beyond a specific choice of prepositions (such as spatial prepositions), towards a systematic classification of preposition senses. In recent years, computational research on prepositions has been enforced, mainly driven by the ACL Special Interest Group on Semantics (ACL-SIGSEM). The SIG has organised a series of workshops on prepositions, and a special issue in the *Computational Linguistics* journal (Baldwin et al., 2009).

Related work across languages includes *The Preposition Project* for English prepositions (Litkowski and Hargraves, 2005), *PrepNet* for French prepositions (Saint-Dizier, 2006), and a German project on the role of preposition senses in determiner omission in prepositional phrases (Kiss et al., 2010). The latter is most closely related to the present work, as it is also aimed at German. Their focus however is on manual classifications and corpus annotation, in contrast to our automatic classification approach.

As in many other languages, German prepositions are notoriously ambiguous, e.g. note the quite distinct senses of the German preposition *nach* in *nach drei Stunden/Berlin/Meinung* ‘after three hours/to Berlin/according to’, referring to a temporal, directional, and accordance meaning. Our gold standard in terms of preposition senses is the German grammar book by Helbig and Buscha (1998). Starting with their class hierarchy, we selected the classes of prepositions that contained more than one preposition. We deleted those prepositions from the classes that appeared less often than 10,000 times in our web corpus containing 880 million words (cf. Section 3.1). This selection process resulted in 12 semantic classes covering between 2 and 27 prepositions each (cf. Table 1). The included prepositions exhibit ambiguity rates of 1 (monosemous) up to 6 (cf. Table 2). Out of the 47 prepositions, 24 are polysemous (51%).

## 3 Cluster Analyses

The pipeline in our framework is as follows.

1. The prepositions are associated with a distributional feature set.
2. The vector space of prepositions is hard-clustered using Self-Organising Maps.

	Class	Size
lokal	‘local’	27
modal	‘modal’	24
temporal	‘temporal’	21
kausal	‘causal’	5
distributiv	‘distributive’	6
final	‘final’	4
urheber	‘creator’	3
konditional	‘conditional’	3
ersatz	‘replacement’	2
restriktiv	‘restrictive’	2
partitiv	‘partitive’	2
kopulativ	‘copulative’	2

Table 1: Preposition classes.

#Senses	#Prepositions
6	1
5	3
4	3
3	11
2	6
1	23

Table 2: Degrees of preposition ambiguity.

3. The hard clustering is transferred to a soft clustering.
4. The cluster analyses are evaluated.

The following subsections describe these steps in more detail. While the larger framework plans to perform this pipeline for various cluster algorithms and many feature sets, the current setup of experiments focuses rather on the methodology towards polysemy detection, and is thus restricted to one algorithm (SOMs) and one feature set (nouns).

### 3.1 Preposition Corpus Features

The distributional features for the German prepositions were induced from the *sdeWaC* corpus (Faaß and Eckart, 2013), a cleaned version of the German web corpus *deWaC* created by the *WaCky* group. The corpus cleaning had focused mainly on removing duplicates from the *deWaC*, and on disregarding sentences that were syntactically ill-formed (relying on a parsability index provided by a standard dependency parser (Schiehlen, 2003)). The *sdeWaC* contains approx. 880 million words.

In this paper, we focus on one specific feature set that is expected to provide salient properties towards preposition meaning, i.e., the nouns that are subcategorised by the prepositions. This dependency information was extracted from a parsed version of the *sdeWaC* using Bohnet’s MATE dependency parser (Bohnet, 2010). So each preposition was associated with a feature vector over its

subcategorised nouns. The overall set of noun features was restricted to the 10,000 nouns from the corpus which co-occurred with the largest number of prepositions.

### 3.2 Hard Clustering

For hard-clustering the German prepositions, we relied on the Self-Organising Maps (SOMs) artificial neural networks provided by the `kohonen` library of the *R Project for Statistical Computing*<sup>1</sup>. We expected SOMs to be especially useful for this task, as they create typology-preserving maps, and should thus provide a suitable model to look into the spatial properties of polysemous vectors. Furthermore, SOMs have successfully been applied to semantic classification before (Ontrup and Ritter, 2001; Kanzaki et al., 2002; Guida, 2007).

We created SOM maps with  $k$  clusters, for  $2 \leq k \leq 47$ , where 47 represents the total number of prepositions. For each  $k$ , we initiated two-dimensional spacings for all possible hexagonal grids. For example, we trained four SOM maps with 30 clusters, using a  $30 \times 1$  grid, a  $15 \times 2$  grid, a  $10 \times 3$  grid, and a  $6 \times 5$  grid. The distance measure used in the maps was *Euclidean Distance*, which is the only option for SOMs in *R*.

### 3.3 Soft Clustering

The soft clustering of the German prepositions was based on the various hard cluster analyses. We performed the *hard*  $\rightarrow$  *soft* clustering transfer in two alternative ways, providing two different types of soft cluster analyses.

**(1) Centroid-based softening:** For each cluster  $c$  within a hard cluster analysis  $C$ , we calculated the mean distance  $prep2cluster(c)$  over all prepositions  $p$  to the cluster centroid  $z_c$ , ignoring any hard assignments in the hard clustering, cf. Equation 1. The individual distances between a preposition  $p$  and a cluster centroid  $z_c$  are denoted as  $d(p, z_c)$ .

$$prep2cluster(c) = \frac{\sum^p d(p, z_c)}{|p|} \quad (1)$$

For the corresponding soft cluster analysis  $S_t(C)$  of a hard cluster analysis  $C$ , a preposition  $p$  was assigned to a cluster  $c$  if the distance  $d(p, z_c)$  was below a threshold  $t \times prep2cluster(c)$ , with  $t = 0.05, 0.1, 0.15, \dots, 0.95$ . For example, if a distance of a preposition  $p$  to a cluster  $c$  was

5, and the mean distance  $prep2cluster(c)$  was 10, then  $p$  would *not* be assigned to  $c$  for  $t = 0.05, 0.1 \dots, 0.5$  but for  $t = 0.6, \dots, 0.95$ . In this way, we created 19 different soft cluster analyses  $S_t(C)$  for each hard clustering  $C$ , one for each  $t$ . With low values of  $t$ , few prepositions (i.e., only those that were very close to the respective cluster centroids) were assigned to the clusters, and the resulting cluster analyses were likely to contain not all of our prepositions, and a low ambiguity rate; with high values of  $t$ , more prepositions were assigned to each of the clusters, and the resulting cluster analyses were likely to contain many of the 47 prepositions, and a high ambiguity rate.

**(2) Preposition-based softening:** For each preposition  $p$  within a hard cluster analysis  $C$ , we calculated the mean distance  $cluster2prep(p)$  over all cluster centroids  $z_c$  to the preposition  $p$ , ignoring any hard assignments in the hard clustering, cf. Equation 2. Again, the individual distances between a preposition  $p$  and a cluster centroid  $z_c$  are denoted as  $d(p, z_c)$ .

$$cluster2prep(p) = \frac{\sum^c d(p, z_c)}{|c|} \quad (2)$$

Similarly to the centroid-based softening, for the corresponding soft cluster analysis  $S_t(C)$  of a hard cluster analysis  $C$ , a preposition  $p$  was assigned to a cluster  $c$  if the distance  $d(p, z_c)$  was below a threshold  $t \times cluster2prep(p)$ , with  $t = 0.05, 0.1, 0.15, \dots, 0.95$ . By relying on the threshold, we again created 19 different soft cluster analyses  $S_t(C)$  for each hard clustering  $C$ , one for each  $t$ . In this case, however, we compared the mean distances of an individual preposition to all cluster centroids, and only performed soft cluster assignments if the preposition was close to a cluster centroid in comparison to its distance to other cluster centroids. With low values of  $t$ , the prepositions were assigned to none or few clusters, and the resulting cluster analyses were likely to contain not all of our prepositions, and a low ambiguity rate; with high values of  $t$ , the prepositions were assigned to many clusters, and the resulting cluster analyses were likely to contain many of the 47 prepositions, and a high ambiguity rate.

## 4 Evaluation

The evaluation metrics play an important role in our work. On the one hand, we created a large number of hard clustering SOMs (i.e., 96 cluster

<sup>1</sup><http://www.r-project.org/>

analyses since we took all possible grids for each  $2 \leq k \leq 47$  into account), and for each hard cluster analysis we created 38 soft cluster analyses (19 centroid-based versions, and 19 preposition-based versions). We thus needed evaluation measures to decide about the quality of a cluster analysis. On the other hand, our methodology relies on evaluation metrics to identify polysemous prepositions, so the measures are crucial to perform this work.

There is a large body of research regarding the question of how to compare and evaluate two cluster analyses. For example, with respect to the specific task of semantic classification, Schulte im Walde (2003), compared a range of evaluation measures. Related work in this area partly adopted the suggested measures, and in addition relied on *Purity* or *Accuracy* (Korhonen et al., 2003; Stevenson and Joanis, 2003). In more general terms, there is an ongoing discussion about cluster comparison, mainly in the field of Machine Learning, but also elsewhere. Recent examples include Meila (2007), Rosenberg and Hirschberg (2007), and Vinh and Bailey (2010). These approaches all concentrate on evaluations relying on the entropy between two cluster analyses, in order to compare them. Entropy is an information-theoretic measure of uncertainty; in our context, entropy measures how uncertain a clustering is, given the information provided by a gold standard, and vice versa.

We decided to make use of two evaluation measures, in order to (i) evaluate and compare our hard and soft cluster analyses, and (ii) detect polysemy. The two measures were expected to provide complementary perspectives on the properties of our cluster analyses, and on the properties of ambiguous prepositions. The following paragraphs describe these measures, and how they were applied.

(1) With the *Silhouette Value* (Kaufman and Rousseeuw, 1990), each cluster is represented by a silhouette displaying which objects lie well within a cluster and which objects are marginal to a cluster. The evaluation appeared specifically suited to our task, as according to our hypotheses, ambiguous prepositions were expected to represent marginals in a cluster analysis, i.e., to be comparably far away from all cluster centroids.

To obtain the silhouette value  $sil$  for an object  $o_i$  within a cluster  $c_A$ , we compared the average distance  $a$  between  $o_i$  and all other objects in  $c_A$  with the average distance  $b$  between  $o_i$  and all objects

in the neighbouring cluster  $c_B$ , cf. Equations 3 to 5. For each object  $o_i$ ,  $-1 \leq sil(o_i) \leq 1$ . If  $sil(o_i)$  is large, the average object distance within the cluster is smaller than the average distance to the objects in the neighbour cluster, so  $o_i$  is well classified. If  $sil(o_i)$  is small, the average object distance within the cluster is larger than the average distance to the objects in the neighbour cluster, so  $o_i$  has been misclassified. The silhouette value was only calculated if cluster  $c_A$  has at least two members, i.e. if it is not a singleton.

$$a(o_i) = \frac{1}{|c_A| - 1} \sum_{o_j \in c_A, o_j \neq o_i} d(o_i, o_j) \quad (3)$$

$$b(o_i) = \min_{c_B \neq c_A} \frac{1}{|c_B|} \sum_{o_j \in c_B} d(o_i, o_j) \quad (4)$$

$$sil(o_i) = \frac{b(o_i) - a(o_i)}{\max\{a(o_i), b(o_i)\}} \quad (5)$$

In addition to providing information about the quality of classification of a single object, the silhouette value can be extended to evaluate the individual clusters and the entire clustering. The *average silhouette width*  $sil(c)$  of a cluster  $c$  is defined as the average silhouette value for all objects within cluster  $c$ , cf. Equation 6, and the *average silhouette width for the clustering*  $C$  with  $k$  clusters  $sil(C_k)$  is defined as the average silhouette value for the individual clusters, cf. Equation 7.

$$sil(c) = \frac{1}{|c|} \sum_{o_i \in c} sil(o_i) \quad (6)$$

$$sil(C_k) = \frac{1}{k} \sum_{i=1}^k sil(c) \quad (7)$$

(2) The *V-Measure* (Rosenberg and Hirschberg, 2007) is an entropy-based cluster evaluation measure. We chose this measure over other entropy-based measures (e.g., *Variance of Information* (VI) (Meila, 2007), and variants suggested by Vinh and Bailey (2010)) because the V-Measure  $v(C)$  balances two desirable properties for a clustering  $C$  of a given dataset: homogeneity (*hom*) and completeness (*com*), cf. Equations 8 to 10.<sup>2</sup>

<sup>2</sup>Note that Equations 8 and 9 differ from those in Rosenberg and Hirschberg (2007) in the denominators of the *else* condition because there were typos in the definitions (personal communication with Andrew Rosenberg).



*Homogeneity* is similar to *purity*, and measures how well the clusters within a cluster analysis map to the classes within a gold standard. If each cluster contains only objects from one gold-standard class, then the entropy is at its minimum,  $H(C|G) = 0$ . This represents a maximally homogeneous clustering. *Completeness* measures how well the classes within a gold-standard map to the clusters within a cluster analysis. If each gold-standard class contains only objects from one cluster, then the entropy is at its minimum,  $H(G|C)$ . This represents a maximally complete clustering, because each gold-standard class is completely contained in a cluster.

$$hom(C) = 1 \text{ if } H(C, G) = 0; \text{ else } 1 - \frac{H(C|G)}{H(C, G)} \quad (8)$$

$$com(C) = 1 \text{ if } H(G, C) = 0; \text{ else } 1 - \frac{H(G|C)}{H(G, C)} \quad (9)$$

$$v(C) = \frac{2 \times hom(C) \times com(C)}{hom(C) + com(C)} \quad (10)$$

There is however a limitation to the V-Measure because it can only be applied to hard classifications which represent an  $N : 1$  relationship between data points and gold-standard classes. This means a given object only belongs to a single class. In our data, this is clearly not the case due to the inherent ambiguity of the prepositions. We thus extended the V-Measure to a fuzzy version *Fuzzy V-Measure (fuzzy v)* that applies to  $N : M$  classifications, where a data point can belong to any number of classes.<sup>3</sup>

As for the original calculation of the entropy values, we must define the joint and conditional probabilities across clusters and gold-standard classes. In Rosenberg and Hirschberg (2007), the joint probability of a cluster  $c$  and a gold-standard class  $g$  was estimated as

$$\hat{p}(c, g) = \frac{a_{cg}}{N}, \quad (11)$$

where  $a_{cg}$  is the number of prepositions shared by  $c$  and  $g$  and  $N$  is the total number of prepositions. Due to the polysemy of prepositions, we must assume that a preposition occurs in multiple classes. Calculating the probability as above would however give too much weight to highly ambiguous prepositions. Our approach is to give each preposition a total mass of 1 and then equally divide its

<sup>3</sup>Thanks to Andrew Rosenberg for valuable discussions.

	$g_1$	$g_2$	$g_3$	$g_4$
$p_1$	0.5	0.5	0	0
$p_2$	0.33	0	0.33	0.33
$p_3$	0	0.5	0.5	0
$p_4$	0	0.5	0	0.5

Table 3: Prepositions in gold standard.

	$g_1$	$g_2$	$g_3$	$g_4$	$\Sigma$
$c_1$	0.83	0.5	0.33	0.33	= 2
$c_2$	0	1	0.5	0.5	= 2

Table 4: Evidence for clusters.

mass across the classes of which it is a member. Thus, Equation 11 becomes:

$$\hat{p}(c, g) = \frac{\mu(c \cap g)}{M}, \quad (12)$$

where  $\mu(c \cap g)$  is the total mass of the prepositions shared by  $c$  and  $g$ , and  $M$  is the total mass of the clustering. Note that  $M$  will only be equal to  $N$  if each preposition belongs to exactly as many clusters as classes.

Example: The prepositions  $p_1, p_3$  and  $p_4$  each belong to two classes, while preposition  $p_2$  belongs to three classes (cf. Table 3). Assuming cluster  $c_1$  contains  $p_1$ , and  $p_2$ , and  $c_2$  contains  $p_3$  and  $p_4$ , the contingency table for the clusters  $c_1$  and  $c_2$  is given as in Table 4. Thus, while both  $c_1$  and  $c_2$  each share two prepositions with the gold-standard classes  $g_1$  and  $g_2$  respectively, the higher ambiguity of  $p_2$  in the first case means there is less evidence for  $c_1$  given  $g_1$  than  $c_2$  given  $g_2$ , namely:  $\hat{p}(c_1|g_1) = .83/2 < 1/2 = \hat{p}(c_2|g_2)$ .

In addition to being applicable to ambiguous data on the side of the classes themselves, our adaptation of the V-Measure also allows for the application to soft clusterings. In this case, the data points may be present in multiple clusters and simply add their respective mass to the cells in the contingency table.

## 5 Detecting Polysemy

This section applies the evaluation measures to our cluster analyses, in order to detect polysemous prepositions, and to identify their spatial properties. Our hypothesis is that polysemous prepositions are outliers, and thus represent either (i) singletons or (ii) marginals of the clusters within a

cluster analysis. We present a series of assumptions regarding this main hypothesis, and check them according to our hard and soft clusterings.

**Singletons represent polysemy.** Our first analysis applies to the hard cluster analyses. The assumption here is that clusters that represent singletons contain polysemous prepositions, because singletons contain objects that do not belong to any of the other clusters. Figure 1 plots the number of polysemous singletons (i.e., those singletons whose only cluster member is a polysemous preposition) against the total number of singletons, for each SOM map. The baseline is provided by 51% of the total number of singletons, as 24 out of our 47 preposition types (51%) are polysemous, so the baseline corresponds to a random assignment of preposition types to singletons.

For SOM maps with up to  $k = 13$  clusters, there is maximally one singleton in the cluster analyses (except for  $k = 4$  and a grid of  $2 \times 2$ , which contains two singletons), so it is difficult to judge about the correctness of our prediction. For  $14 \leq k \leq 26$ , in most cases the number of polysemous singletons clearly outperforms the baseline. For  $k = 22$  with a grid of  $22 \times 1$  and  $k = 26$  with a grid of  $13 \times 2$ , the difference to the baseline is even significant ( $\chi^2$ ,  $p < 0.1$ ). For  $k > 27$ , the number of polysemous singletons outperforms the baseline in fewer cases than for smaller  $k$ . In sum, our prediction that singletons represent polysemy holds for a restricted subset of our SOM maps, most strongly for  $22 \leq k \leq 26$ .

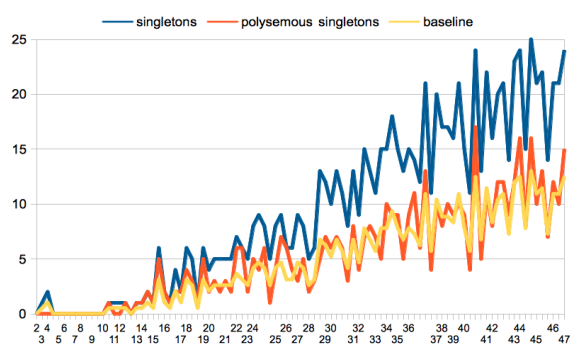


Figure 1: Number of (ambiguous) singletons.

**Polysemous prepositions are misclassified.** Our second analysis also applies to the hard cluster analyses. Figure 2 exploits the Silhouette Value to predict polysemous prepositions. Since prepositions with several senses are expected to represent marginals in a cluster analysis, they

should be comparably far away from all cluster centroids, and thus their silhouette value  $sil$  should be low, i.e., misclassify them. Figure 2 plots the correlation values of Kendall's  $\tau$ -b<sup>4</sup> between the silhouette value  $sil(p)$  and the ambiguity rate  $amb(p)$  as defined by the gold standard, across all hard cluster analyses. According to our hypothesis,  $\tau$  should be negative: the higher the ambiguity rate, the lower the silhouette value.

The plot demonstrates that our assumption is only partly correct: There are cluster analyses where we find a weak negative correlation, but most clusterings do not exhibit a noticeable correlation, and some clusterings even have a moderate positive correlation. For  $k = 24$  with a grid of  $24 \times 1$  and  $k = 27$  with a grid of  $27 \times 1$ , we however find cluster analyses with a moderate negative correlation,  $\tau = -0.30$  and  $\tau = -0.32$ .

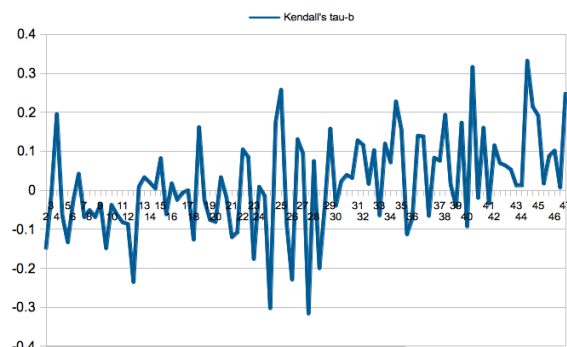


Figure 2: Correlation between  $sil(p)$  and  $amb(p)$ .

**General evaluation of soft clusterings.** Before we move on to exploring a further hypothesis regarding polysemous prepositions, we present a general evaluation of our two types of softening approaches. Figures 3 and 4 plot the *homogeneity*, *completeness* and *fuzzy v* scores after applying centroid-based and preposition-based softening to  $k$  hard clusters, respectively. The soft cluster analyses depend on the threshold  $t$  that controls the assignment of prepositions to clusters. We chose  $t = 0.7$  as a medium threshold for the two figures. Since the various  $k$  cause strong differences in the coverage of the preposition types in the soft cluster analyses, we also plot the *coverage*, and the harmonic mean of *fuzzy v* and *coverage*.

The best *fuzzy v* scores for the centroid-based soft clusters were obtained with  $k = 16$  and a  $8 \times 2$  grid (0.380),  $k = 12$  with a  $6 \times 2$  grid (0.379) and  $k = 10$  with a  $10 \times 1$  grid (0.377).

<sup>4</sup>Kendall's  $\tau$ -b is a measure of association based on concordant and discordant pairs, adjusted for the number of ties.

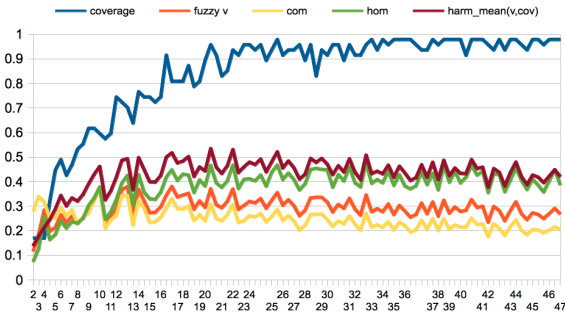


Figure 3: Centroid-based softening: evaluation.

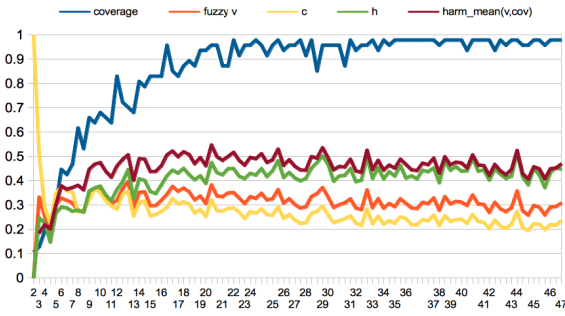


Figure 4: Preposition-based softening: evaluation.

If we take the coverage into account, the best results were obtained with  $k = 20$  with a  $20 \times 1$  grid (0.534),  $k = 22$  with a  $11 \times 2$  grid (0.530) and  $k = 25$  with a  $5 \times 5$  grid (0.521). For the preposition-based soft clusters the respective *fuzzy v* scores were  $k = 12$  with a  $6 \times 2$  grid (0.396),  $k = 16$  with a  $8 \times 2$  grid (0.376) and  $k = 29$  with a  $29 \times 1$  grid (0.372); taking coverage into account, the respective scores were  $k = 20$  with a  $20 \times 1$  grid (0.547),  $k = 29$  with a  $29 \times 1$  grid (0.536) and  $k = 25$  with a  $5 \times 5$  grid (0.530). In sum, the best *fuzzy v* scores for both types of soft cluster analyses were in most cases obtained for  $k$  being similar to the number of gold standard classes. Taking coverage into account, the best results were obtained for cluster analyses with  $20 \leq k \leq 29$ .

A threshold of  $t = 0.7$  seemed appropriate for our descriptions, since lower and also higher values of  $t$  resulted in less clear preferences for  $k$ , and the threshold appeared like a useful compromise between low coverage in assigning prepositions to clusters, and highly ambiguous clusters.

**Correlation of cluster membership rate with ambiguity rate.** This final analysis investigates the relationship between the cluster membership rate of a preposition and its ambiguity rate. Our assumption is that the more clusters a specific preposition is assigned to, the more ambiguous it is. As

basis for this analysis we used both the centroid-based and the preposition-based soft clusters, with varying  $t$ . Figures 5 and 6 present the correlation results, again relying on *Kendall's tau-b*. For presentation reasons, we restrict the plots to  $10 \leq k \leq 30$  with grid shapes  $k \times 1$  only, and  $t = 0.6, 0.7, 0.8, 0.9$ .

Both plots demonstrate that the highest threshold  $t = 0.9$  corresponding to highly ambiguous cluster analyses exhibits the best correlations with the ambiguity rates of the prepositions. For the centroid-based softening, this is true for  $12 \leq k \leq 20$ , for the preposition-based softening, this is true for all but two values of  $k$ . For lower thresholds, it seems that  $t = 0.8 > t = 0.7 > t = 0.6$ , but the differences are not at all clear but rather vary depending on  $k$ . Overall, we reached moderate correlation values, the best correlation being  $\tau = 0.45$ . Interestingly, the best correlation values in the two types of softening approaches were obtained for similar values of  $k$ , and with  $k$  being very similar to the number of gold standard classes (12): the prediction of the centroid-based softening was best with  $k = 13$  and  $k = 12$  ( $\tau = 0.453$  and  $\tau = 0.449$ , respectively), and the prediction of the preposition-based softening was best with  $k = 12$  and  $k = 14$  ( $\tau = 0.439$  and  $\tau = 0.368$ ).

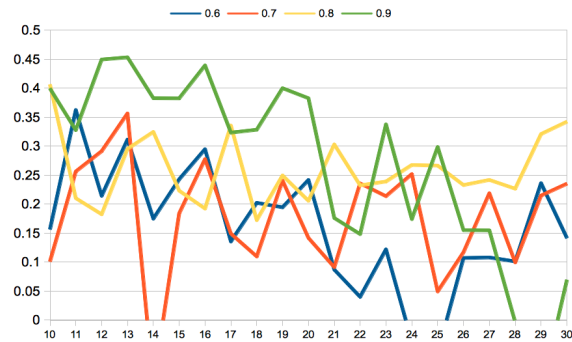


Figure 5: Centroid-based softening: ambiguity.

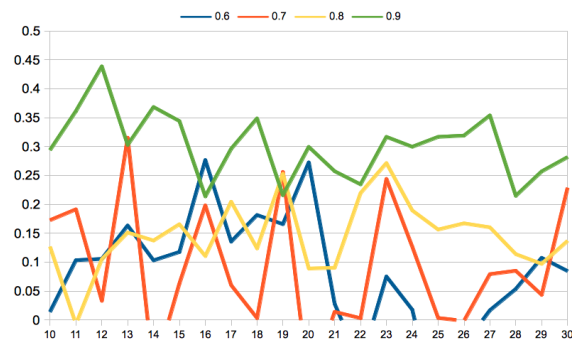


Figure 6: Preposition-based softening: ambiguity.

## 6 Discussion

In the previous section, we performed a series of analyses to investigate the spatial properties of polysemous prepositions in vector space models. Our main hypothesis is that polysemous prepositions are outliers, and thus represent either (i) singletons or (ii) marginals of the clusters within a cluster analysis. Concerning option (i), we showed that for specific values of  $k$ , there were significantly more polysemous prepositions in the singletons of the hard clusterings than there would be by chance. The relationship did not hold across  $k$ , however. Concerning option (ii), we performed two analyses. First, we checked whether the silhouette value of a preposition in a hard clustering correlated with its ambiguity rate, based on the assumption that the silhouette value identifies cluster marginals. Again, we found a strong correlation for specific values of  $k$ , but not across  $k$ . Second, relying on the soft clusterings we checked whether the cluster membership rate of a preposition correlated with its ambiguity rate: Especially in highly ambiguous cluster analyses there were strong correlations in both types of soft clusterings, for  $k$  similar to the number of gold standard classes.

In sum, our analyses confirmed our hypothesis, but (a) with regard to specific  $k$  only, and (b) the  $k$  varied across the analyses. This might partly be due to our clustering approaches (SOMs for hard clustering, and our two versions of softening approaches), so we are currently experimenting with alternatives. Furthermore, the *fuzzy v* measure that we developed in order to evaluate soft clusterings still seems to provide sub-optimal evidence of clustering quality: The magnitude of the score depends on the threshold, so it is difficult to decide which threshold performed best.

On the other hand, several of our analyses pointed towards similar numbers for an optimal  $k$ , and these optimal  $k$  values were reasonable, as they were close to the number of gold standard classes. Last but not least, we looked into a range of clusterings that performed well according to our *fuzzy v*, and it turned out that within a certain magnitude of  $k$ , the clusterings were very similar to each other, with similar strengths and weaknesses. We thus conclude this paper with a qualitative analysis of the centroid-based soft clustering with  $k = 16$  and a  $8 \times 2$  grid, the best clustering according to the general evaluation.

The clustering actually contained only 15 clus-

ters (so one cluster was an empty cluster). Three of the clusters were singletons, one with a 3-way ambiguous preposition (*nach*: local, modal, temporal), one with a 2-way ambiguous preposition (*unter*: local, modal), and one with a monosemous preposition (*samt*: modal). From the remaining 12 clusters, 8 could unambiguously be assigned a major sense according to the gold standard classes, and 4 clusters contained prepositions from various gold standard classes.

Overall, we found 27 local preposition senses, 24 modal senses, 21 temporal senses, 5 causal and 3 replacement senses. The minor senses (according to the sizes of the gold standard classes), i.e., final, creator, distributive, partitive, conditional, copulative and restrictive, were not found in the clustering. So there was a clear bias towards the assignment of majority senses. This bias might well be due to the very different sizes of the gold standard classes, so in future work we will experiment with sub-classifications of the large classes.

## 7 Conclusion

In this paper, we presented a methodology to identify polysemous German prepositions by exploring their vector spatial properties in hard and soft clusterings. The analyses demonstrated that – when looking at clusterings with a similar or slightly larger number of clusters than the gold standard – (a) singletons have a tendency to contain polysemous prepositions; and (b) misclassification and cluster membership rate exhibit a moderate correlation with ambiguity rate.

## Acknowledgements

The research presented in this paper was funded by the DFG Collaborative Research Centre SFB 732 (Sylvia Springorum, Jason Utt), and the DFG Heisenberg Fellowship SCHU-2580/1-1 (Sabine Schulte im Walde).

## References

- Timothy Baldwin, Valia Kordoni, and Aline Villavicencio, editors. 2009. *Computational Linguistics, Volume 35, Number 2, June 2009 - Special Issue on Prepositions*, volume 35. MIT Press.
- 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.
- Gemma Boleda, Sebastian Padó, and Jason Utt. 2012a. Regular Polysemy: A Distributional Model. In

- Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*, pages 151–160, Montréal, Canada.
- Gemma Boleda, Sabine Schulte im Walde, and Toni Badia. 2012b. Modelling Regular Polysemy: A Study on the Semantic Classification of Catalan Adjectives. *Computational Linguistics*, 38(3):575–616.
- Katrin Erk and Sebastian Padó. 2010. Exemplar-based Models for Word Meaning in Context. In *Proceedings of the ACL Conference Short Papers*, pages 92–97, Uppsala, Sweden.
- Katrin Erk. 2009. Representing Words in Regions in Vector Space. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, pages 57–65, Boulder, Colorado.
- Katrin Erk. 2012. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653.
- Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – a Corpus of Parsable Sentences from the Web. In *Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology*, Darmstadt, Germany. To appear.
- Annamaria Guida. 2007. The Representation of Verb Meaning within Lexical Semantic Memory: Evidence from Word Associations. Master’s thesis, Università degli studi di Pisa.
- Gerhard Helbig and Joachim Buscha. 1998. *Deutsche Grammatik*. Langenscheidt – Verlag Enzyklopädie, 18th edition.
- Kyoko Kanzaki, Qing Ma, Masaki Murata, and Hitoshi Isahara. 2002. Classification of Adjectival and Non-Adjectival Nouns based on their Semantic Behaviour by using a Self-Organizing Semantic Map. In *Proceedings of the COLING Workshop SEMANET: Building and Using Semantic Networks*, Taipei, Taiwan.
- Leonard Kaufman and Peter J. Rousseeuw. 1990. *Finding Groups in Data – An Introduction to Cluster Analysis*. Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York.
- Tibor Kiss, Katja Keßelmeier, Antje Müller, Claudia Roch, Tobias Stadtfeld, and Jan Strunk. 2010. A Logistic Regression Model of Determiner Omission in PPs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 561–569, Beijing, China.
- Teuvo Kohonen. 2001. *Self-Organizing Maps*. Springer, Berlin, 3rd edition.
- Anna Korhonen, Yuval Krymowolowski, and Zvika Marx. 2003. Clustering Polysemic Subcategorization Frame Distributions Semantically. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 64–71, Sapporo, Japan.
- Kenneth C. Litkowski and Orin Hargraves. 2005. The Preposition Project. In *Proceedings of the 2nd ACL-SIGSEM Workshop on The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, pages 171–179, Colchester, England.
- Kevin Lund and Curt Burgess. 1996. Producing High-Dimensional Semantic Spaces from Lexical Co-Occurrence. *Behavior Research Methods, Instruments, and Computers*, 28(2):203–208.
- Marina Meila. 2007. Comparing Clusterings - An Information-based Distance. *Journal of Multivariate Analysis*, 98(5):873–895.
- Jörg Ontrup and Helge J. Ritter. 2001. Hyperbolic Self-Organizing Maps for Semantic Navigation. *Advances in Neural Information Processing Systems*.
- Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A Conditional Entropy-based External Cluster Evaluation Measure. In *Proceedings of the joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, Prague, Czech Republic.
- Patrick Saint-Dizier. 2006. PrepNet: a Multilingual Lexical Description of Prepositions. In *Proceedings of the 5th Conference on Language Resources and Evaluation*, pages 1021–1026, Genoa, Italy.
- Michael Schiehlen. 2003. A Cascaded Finite-State Parser for German. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 163–166, Budapest, Hungary.
- Sabine Schulte im Walde. 2003. *Experiments on the Automatic Induction of German Semantic Verb Classes*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart. Published as AIMS Report 9(2).
- Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123. Special Issue on Word Sense Disambiguation.
- Suzanne Stevenson and Eric Joanis. 2003. Semi-supervised Verb Class Discovery Using Noisy Features. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 71–78, Edmonton, Canada.
- Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.
- Nguyen Xuan Vinh and James Bailey. 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research*, 11:2837–2854.

# Generalized Abbreviation Prediction with Negative Full Forms and Its Application on Improving Chinese Web Search

Xu Sun<sup>†</sup>, Wenjie Li<sup>‡</sup>, Fanqi Meng<sup>‡</sup>, Houfeng Wang<sup>†</sup>,

<sup>†</sup>Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, China

<sup>‡</sup>Department of Computing, The Hong Kong Polytechnic University

xusun@pku.edu.cn cswjli@comp.polyu.edu.hk mengfanqi928@163.com wanghf@pku.edu.cn

## Abstract

In Chinese abbreviation prediction, prior studies are limited on positive full forms. This lab assumption is problematic in real-world applications, which have a large portion of negative full forms (NFFs). We propose solutions to solve this problem of generalized abbreviation prediction. Experiments show that the proposed unified method outperforms baselines, with the full-match accuracy of 79.4%. Moreover, we apply generalized abbreviation prediction for improving web search quality. Experimental results on web search demonstrate that our method can significantly improve the search results, with the search F-score increasing from 35.9% to 64.9%. To our knowledge, this is the first study on generalized abbreviation prediction and its application on web search.

## 1 Introduction

Abbreviations increase the ambiguity in a text. Associating abbreviations with their fully expanded forms is important in various natural language processing applications (Pakhomov, 2002; Yu et al., 2006; HaCohen-Kerner et al., 2008). Chinese abbreviations represent fully expanded forms (e.g., the left side of Figure 1) through the use of shortened forms (e.g., the right side of Figure 1). Chinese abbreviations are derived via a generative lexical process. Although native speakers may possess intuitions of the generative process, it cannot be adequately explained by any linguistic theory (Chang and Lai, 2004; Chang and Teng, 2007).

Abbreviation prediction (i.e., predicting abbreviation of a given full form) is important in Chinese natural language processing applications. For example, it is helpful for information retrieval if we can estimate the abbreviation of a query. For

Full Form	Abbr.
欧洲经济与货币联盟	→ 欧盟
European Economic and Monetary Union	

Figure 1: An example of abbreviation prediction.

the data of one month's People's Daily, only 17% of the documents contain the full form in Figure 1, while more than 70% of the articles contain only the abbreviation in Figure 1. It is expected that abbreviation prediction can improve the recall in information retrieval. In addition, (Yang et al., 2009b) in speech studies showed that Chinese abbreviation prediction can improve voice-based search quality.

The study of Chinese abbreviation prediction is still in an early stage. Chinese abbreviation prediction is quite different from English ones, because of its specific characteristics (Sun et al., 2008; Huang et al., 1994; Chang and Teng, 2007; Yang et al., 2009b; Yang et al., 2009a). For example, Chinese abbreviations are not necessarily from the initials of words. They frequently take non-initial characters from the words in the full form. In addition, the Chinese full form does not have word boundaries.

To our knowledge, all of the prior studies of Chinese abbreviation prediction (Sun et al., 2008; Sun et al., 2009; Sun et al., 2013; Huang et al., 1994; Chang and Teng, 2007; Yang et al., 2009b; Yang et al., 2009a) have focused on positive full forms with valid abbreviations. This implicit lab assumption is quite limited in real-world applications, because real-world Chinese full forms contain a large portion of negative full forms (NFFs), which have no abbreviation at all. Abbreviation prediction becomes more difficult by considering NFFs, because of the strong noise. This difficulty is one of the reason of the lab setting on considering only positive full forms. Another reason is

probably the difficulty of data collection. To our knowledge, there is no existing collection of abbreviation prediction data with NFFs.

We aim at solving this abbreviation prediction problem with generalized assumption (hereinafter *generalized abbreviation prediction*). We manually collected a large dataset for this study, which contains 10,786 entries including NFFs. To deal with the strong noise from NFFs, we propose a variety of solutions. We also apply generalized abbreviation prediction for web search and we show that it can significantly improve web search quality.

## 2 Proposed Method

### 2.1 Preprocessing

The preprocessing step includes word segmentation and part-of-speech tagging for the input abbreviation prediction full forms. The word segmentation and part-of-speech tagging is done via the tool ICTCLAS<sup>1</sup>.

### 2.2 Abbreviation Prediction

#### 2.2.1 Simple Heuristic System

The simple heuristic system means always choosing initial characters of words in the segmented full form. This is because the most natural abbreviating heuristic is to produce the first character of each word in the original full form. This is just the simplest baseline.

#### 2.2.2 Unified System

We present a unified system for generalized abbreviation prediction with NFFs. The unified system can conduct the abbreviation prediction with a single step. We cast abbreviation prediction as a sequential labeling task. Following (Sun et al., 2013), each character in the full form is tagged with a label,  $y \in \{P, S\}$ , where the label  $P$  produces the current character and the label  $S$  skips the current character.

As for NFFs, we need a special encoding of labeling  $E$  to represent “no valid abbreviation”. Since there is no prior work, we need to study this “no valid abbreviation” issue. Given a full form  $F$ , its valid abbreviation  $A$  should have the character number constraints:  $0 < |A| < |F|$ . On the other hand, we can assume that a negative full form has an “invalid abbreviation”  $A$  with  $|A| = 0$

<sup>1</sup><http://ictclas.org/>

or  $|A| = |F|$ . Those two kinds of interpretations actually represent two different answers to the question “why some full forms do not have valid abbreviations”:

- Assumption-1 with  $|A| = 0$ : It assumes that a negative full form  $F$  is “abbreviated” to nothing, i.e., with the abbreviation  $A = NULL$ .
- Assumption-2 with  $|A| = |F|$ : It assumes that a negative full form  $F$  is “abbreviated” to itself, i.e., with the abbreviation  $A = F$ .

We want to find out which assumption leads to better performance.

With those interpretations, invalid abbreviations are treated as special forms of abbreviations, thus positive and negative full forms can be modeled in a unified framework via sequential labeling. For simplicity, we use the well-known conditional random fields (CRFs) (Lafferty et al., 2001) for sequential labeling. Assuming a feature function that maps a pair of observation sequence  $\mathbf{x}$  (characters of a full form) and label sequence  $\mathbf{y}$  (label encoding based on abbreviations) to a feature vector  $\mathbf{f}$ , the probability function is defined as follows:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \frac{\exp[\mathbf{w}^\top \mathbf{f}(\mathbf{y}, \mathbf{x})]}{\sum_{\forall \mathbf{y}'} \exp[\mathbf{w}^\top \mathbf{f}(\mathbf{y}', \mathbf{x})]}, \quad (1)$$

where  $\mathbf{w}$  is a parameter vector.

Given a training set consisting of  $n$  labeled sequences,  $(\mathbf{x}_i, \mathbf{y}_i)$ , for  $i = 1 \dots n$ , parameter estimation is performed by maximizing the objective function,

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^n \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}) - R(\mathbf{w}). \quad (2)$$

The second term is a regularizer, typically an  $L_2$  (Gaussian) norm,  $R(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2\sigma^2}$ .

We use features as follows:

- Character feature This feature records the input characters  $x_{i-1}$ ,  $x_i$  and  $x_{i+1}$ .
- Character bi-gram The character bi-grams starting at  $(i-2) \dots i$ .
- Numeral Whether or not the  $x_i$  is a numeral.



- Organization name suffix Whether or not the  $x_i$  is a suffix of traditional Chinese organization names.
- Location name suffix Whether or not the  $x_i$  is a suffix of traditional Chinese location names.
- Word segmentation information After the word segmentation step, whether or not the  $x_i$  is the beginning character of a word.
- Part-of-speech information The part-of-speech tag information of  $x_i$ .

$i$  denotes the current position for extracting features.

The character unigram and bi-gram feature is to capture character-based information in the abbreviating process. For example, some special characters are more likely to be chosen in abbreviating. The named entity suffix features are used because a named entity suffix character is more likely to be chosen in abbreviating. The word segmentation information is also important because the beginning character of a word is more likely to be chosen in abbreviating.

### 2.3 Label Encoding with Global Information

As a common practice to reduce complexity, only local information based on Markov assumption is used for sequential labeling. Nevertheless, the Chinese abbreviation generation process is highly dependent on global information. An example of global information is the number of characters of the generated abbreviations.

For better performance, we try to model global information to make the system be “aware” of the number of characters being generated. We use a simple, effective, and tractable solution for modeling global information in abbreviation prediction: label encoding with global information (GI) (Sun et al., 2013).

In this approach, the label  $y_i$  at position  $i$  will be encoded with the global information of its previous labels,  $y_1, y_2, \dots, y_{i-1}$ . Note that, while directly increasing the Markov order is untractable, the GI label encoding is tractable. More detailed description of the GI method is in (Sun et al., 2013).

Category	Portion (%)
Noun Phrase	52.01%
Verb Phrase	13.72%
Organization Name	26.84%
Location Name	5.28%
Person Name	0.32%
Others	1.80%

Table 1: Distribution of the full forms in the data.

### 2.4 Abbreviation Prediction for Web Search

Abbreviation prediction should be helpful for information search, but we find there is almost no prior work on this. It is probably because the traditional abbreviation prediction is not so applicable in real-world data, which includes lots of NFFs. Since we have solved this problem via generalized abbreviation prediction, we hope to apply generalized abbreviation prediction on improving information search.

In particular, we apply generalized abbreviation prediction for “query expansion” in Chinese web search. In this method, the original queries are treated as full forms (with NFFs) for generating abbreviations. Given a query as an input, the generalized abbreviation prediction system outputs an abbreviation candidate or a NULL string. If the output is an empty string, it means the query is an NFF. Finally, the derived abbreviations, together with the original query terms, are used for web search, and their search results (web pages) are simply added together. For the negative full forms, only the original full forms are used for the web search. The simple architecture of the query expansion system is summarized in Figure 2.

In addition, to make clear the role of the predicted abbreviations in web search, we can remove the full form information in web search and check the difference. In this way, the method turns to a “query alternation” method, which uses the predicted abbreviation to *replace* the positive full form for the web search. For a negative full form, the query alternation acts the same like the query expansion.

## 3 Experiments

Here we describe our collected data for generalized abbreviation prediction. First, we extract long phrases and terms from Chinese natural language processing corpora, including People’s Daily cor-



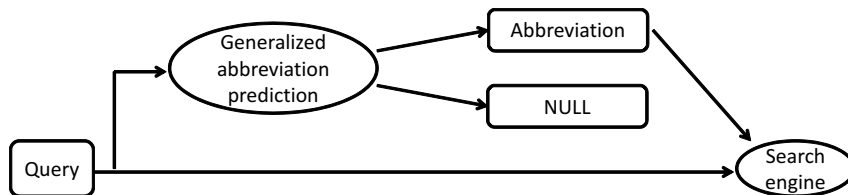


Figure 2: Generalized abbreviation prediction for improving web search.

Positive/Negative Full Forms	Abbreviation
磷酸氢二钠	X
君主专制制	X
珠穆朗玛峰	珠峰
天公不作美	X
中国社会科学院	中国社科院
新时期的总任务	X
自由民主党	自民党
车辆发动机	X
复员退伍军人安置办公室	复退办
车尔尼雪夫斯基	X
持谨慎态度	X
土产日用品杂品公司	土杂公司
一叶蔽目不见泰山	X
打击黑势力扫除恶势力	打黑扫恶

Figure 3: Samples of the collected data with NFFs. The “X” means no valid abbreviation.

pora<sup>2</sup> and SIGHAN word segmentation corpora<sup>3</sup>. Then, we classify the collected phrases and terms into positive and negative full forms. For the negative full forms, no further annotation is required. For the positive full forms, their abbreviations are annotated.

We build a dataset containing 10,786 full forms, including 8,015 positive full forms and 2,771 negative full forms. Samples of the data are shown in Figure 3. The dataset is made up of phrases and terms, including noun phrases, verb phrases, organization names, location names, and so on. The distribution is shown in Table 1. For experiments, we randomly sampled 8,629 samples (80% of the full dataset) for training and 2,157 (20% of the full dataset) for testing.

For experiments on web search, we simply use the 2,157 testing samples as query terms for web

search. The evaluation is on the news domain<sup>4</sup> of the well-known web search engine “baidu.com”<sup>5</sup>. The Baidu news search engine has two alternative options: “title search” and “full content search”. Since abbreviations are more common in news titles, we adopt the option of title search.

### 3.1 Experimental Settings

For evaluating abbreviation prediction quality, the systems are evaluated using the following two metrics:

- **All-match accuracy (All-Acc):** The number of correct outputs (i.e., label strings) generated by the system divided by the total number of full forms in the test set.<sup>6</sup>
- **Character accuracy (Char-Acc):** The number of correct labels (i.e., a classification on a character) generated by the system divided by the total number of characters in the test set.

For evaluating web search quality based on a given query, the following metrics are used:

- **Precision  $P$ :** The number of correct search results returned by the query divided by the total number of search results returned by the query.
- **Recall  $R$ :** The number of correct search results returned by the query divided by the total number of existing correct search results based on the query.
- **F-Score  $F$ :**  $F = 2PR / (P + R)$ .

<sup>4</sup>We choose news domain because abbreviations are mainly from named entities, and named entities are important in news domain.

<sup>5</sup><http://news.baidu.com>

<sup>6</sup>There is only one label string for a full form, and a label string corresponds to a unique abbreviation candidate. A label string is deemed as correct if and only if *all* of the labels are correct.

<sup>2</sup>[http://ic1.pku.edu.cn/ic1\\_res](http://ic1.pku.edu.cn/ic1_res)

<sup>3</sup><http://www.sighan.org/bakeoff2005>

Method	Discriminate Acc (%)	Overall All-Acc	Overall Char-Acc
Heuristic System	73.20	25.77	65.79
Unified-Assum.1 (Perc)	87.48	54.89	87.02
Unified-Assum.1 (MEMM)	86.97	50.16	85.92
Unified-Assum.1 (CRF-ADF)	87.80	56.69	87.20
Unified-Assum.1-GI (Perc)	91.93	75.42	90.23
Unified-Assum.1-GI (MEMM)	88.59	70.32	88.21
Unified-Assum.1-GI (CRF-ADF)	91.05	<b>79.46</b>	<b>91.61</b>
Unified-Assum.2 (Perc)	86.83	55.86	82.20
Unified-Assum.2 (MEMM)	87.52	56.18	82.27
Unified-Assum.2 (CRF-ADF)	87.11	56.97	82.54
Unified-Assum.2-GI (Perc)	90.35	71.85	88.04
Unified-Assum.2-GI (MEMM)	87.99	63.74	83.77
Unified-Assum.2-GI (CRF-ADF)	90.77	74.78	89.19

Table 2: Results on comparing different methods on generalized abbreviation prediction. *Assum.1* and *Assum.2* represent the two assumptions on NFFs discussed in Section 2.2.2. *GI* means the integration with global information. As we can see, the *Unified-Assum.1-GI (CRF)* system has the best performance.

For evaluating web search quality based on a set of queries, we use the macro-averaging and micro-averaging of the precision, recall, and F-score based on a single query. Hence, we finally have six metrics: macro-precision, macro-recall, macro-F-score, micro-precision, micro-recall, and micro-F-score. We use the novel training method, adaptive online gradient descent based on feature frequency information (ADF) (Sun et al., 2012), for fast and accurate training of the CRF model.

To study the performance of other machine learning models, we also implement on other well-known sequential labeling models, including maximum entropy Markov models (MEMMs) (McCallum et al., 2000) and averaged perceptrons (Perc) (Collins, 2002).

### 3.2 Results on Abbreviation Prediction

The experimental results are shown in Table 2. In the table, the *overall accuracy* is most important and it means the final accuracy achieved by the systems in generalized abbreviation prediction with NFFs. For the completeness of experimental information, we also show the *discriminate accuracy*. The *discriminate accuracy* checks the accuracy of discriminating positive and negative full forms, without comparing the generated abbreviations with the gold-standard abbreviations.

As we can see from Table 2, first, the best system is the system *Unified-Assum.1-GI (CRF)*. Results demonstrate that incorporating global information can always improve the accuracy for

the unified methods. Second, the unified system with *assumption-1* has better accuracy than the one with *assumption-2*. This result suggests that *assumption-1* works better in practice. It is interesting that *assumption-1* is more useful. A probable reason is that those negative full forms have no similar patterns with the real abbreviations. For example, the number of characters in NFFs is very different compared with that of real abbreviations. Also, the NFFs contain more formal word units. Real abbreviations contain much less word units. Thus, *assumption-2* will have the inconsistency problem between abbreviations generated from NFFs and real abbreviations. As a result, *assumption-2* works worse than *assumption-1* which gives no abbreviations. Finally, the CRF model outperforms the MEMM and averaged perceptron models. To summarize, the unified system with *assumption-1*, global information, and CRF model has the best performance.

### 3.3 Results on Web Search

We use the 2,157 testing samples as query terms for web search. We test the original query terms, the query alternation, and the query expansion methods. Some search results actually do not match the query. For example, given a query *abc*, some search results do not contain *abc*, but with the expression “*ab...c*” or “*a...bc*”, where “*...*” means other characters. In this case, the search results are incorrect. Since the number of the search results is massive, we need to evaluate the web

Method	Micro Prec	Micro Rec	Micro F1	Macro Prec	Macro Rec	Macro F1
Original query	48.51	18.14	26.41	47.84	28.76	35.92
Query alternation	47.73	54.04	50.69	62.84	61.12	61.97
Query expansion	47.93	72.18	<b>57.60</b>	53.70	82.02	<b>64.90</b>

Table 3: Results on comparing different methods on web search quality.

Method	Micro Prec	Micro Rec	Micro F1	Macro Prec	Macro Rec	Macro F1
Original query	48.51	18.14	26.41	47.84	28.76	35.92
Query alternation (gold-standard)	72.31	81.86	76.79	83.07	79.11	<b>81.04</b>
Query expansion (gold-standard)	66.40	100.00	<b>79.81</b>	65.56	100.00	79.19

Table 4: Results on comparing different methods on web search quality.

search quality in an efficient way. The correctness of the search results is evaluated by an automatic postprocessing scoring system, which crawls the search results from the `baidu.com` site. Then, the system runs text matching analysis to check if the search query matches the retrieved web pages.

In traditional web search studies, many queries are phrases (e.g., NP+VP) with ambiguous senses. In this case, improving search precision via a contextual information is important. However, for abbreviation processing, most abbreviations are from named entities (the data contains phrases but most of them are NFFs), and the major problem of named entities is variational expressions. In this case, the search recall is more important. To calculate the recall rate in web search, we need to estimate the total number of correct web pages  $N$  relating to a query  $Q$  and its abbreviation  $A$ . We can estimate  $N$  via summing up the correct web pages of  $Q$  and the correct web pages of the gold-standard abbreviation  $A$ .

The precision, recall, F-scores are shown in Table 3. As we can see, the query expansion method based on generalized abbreviation prediction achieves significantly better F-scores on web search quality than using the original queries. We find the major improvement is from the recalls. As expected, the query alternation has lower recall rate than the query expansion method, because the full form information is removed. Nevertheless, the query alternation method is also better than using the original queries. This result emphasizes that the abbreviations are helpful in title-based news search.

Finally, we check the “up-bound” of the performance of generalized abbreviation prediction for web search. The up-bound is achieved by the 100% correct “gold-standard” system, in which

gold-standard abbreviations labeled in the data are used. The results are shown in Table 4. As we can see, the up-bound of the micro-F-score and the macro-F-score is 79.81% and 81.04%, respectively. Thus, the web search quality of automatic generalized abbreviation prediction still has a large space to be improved, possibly via a larger training data set in the future.

## 4 Conclusions and Future Work

This paper is dedicated on generalized abbreviation prediction and its application on improving web search. Experiments demonstrate that the unified system based on global information outperforms the baselines. Experiments also demonstrate that generalized abbreviation prediction can improve web search qualities. As future work, we try to improve the performance via collecting more training data or via semi-supervised learning methods.

## Acknowledgments

This work was supported by the Hong Kong Polytechnic University Internal Grant (4-ZZD5), Major National Social Science Fund of China (No.12&ZD227), National High Technology Research and Development Program of China (863 Program) (No.2012AA011101), and National Natural Science Foundation of China (No.91024009).

## References

Jing-Shin Chang and Yu-Tso Lai. 2004. A preliminary study on probabilistic models for chinese abbreviations. In *Proceedings of the Third SIGHAN Workshop on Chinese Language Learning*, pages 9–16.

- Jing-Shin Chang and Wei-Lun Teng. 2007. Mining atomic chinese abbreviation pairs: A probabilistic model for single character word recovery. *Language Resources and Evaluation*, 40:367–374.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP'02*, pages 1–8.
- Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2008. Combined one sense disambiguation of abbreviations. In *Proceedings of ACL'08: HLT, Short Papers*, pages 61–64, June.
- C.R. Huang, W.M. Hong, , and K.J. Chen. 1994. Suoxie: An information based lexical rule of abbreviation. In *Proceedings of the Second Pacific Asia Conference on Formal and Computational Linguistics II*, pages 49–52.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pages 282–289.
- Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA.
- Serguei Pakhomov. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of ACL'02*, pages 160–167.
- Xu Sun, Houfeng Wang, and Bo Wang. 2008. Predicting chinese abbreviations from definitions: An empirical learning approach using support vector regression. *Journal of Computer Science and Technology*, 23(4):602–611.
- Xu Sun, Naoaki Okazaki, and Jun'ichi Tsujii. 2009. Robust approach to abbreviating terms: A discriminative latent variable model with global information. In *Proceedings of the ACL'09*, pages 905–913, Suntec, Singapore, August.
- Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of ACL'12*, pages 253–262.
- Xu Sun, Naoaki Okazaki, Jun'ichi Tsujii, and Houfeng Wang. 2013. Learning abbreviations from chinese and english terms by modeling non-local information. *ACM Trans. Asian Lang. Inf. Process.*, 12(2):5.
- Dong Yang, Yi-Cheng Pan, and Sadaoki Furui. 2009a. Automatic chinese abbreviation generation using conditional random field. In *Proceedings of HLT-NAACL'09 (Short Papers)*, pages 273–276.
- Dong Yang, Yi-Cheng Pan, and Sadaoki Furui. 2009b. Vocabulary expansion through automatic abbreviation generation for chinese voice search. In *Proceedings of INTERSPEECH'09*, pages 728–731. ISCA.
- Hong Yu, Won Kim, Vasileios Hatzivassiloglou, and John Wilbur. 2006. A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations. *ACM Transactions on Information Systems (TOIS)*, 24(3):380–404.

# Prosody-Based Unsupervised Speech Summarization with Two-Layer Mutually Reinforced Random Walk

Sujay Kumar Jauhar, Yun-Nung Chen, and Florian Metze

School of Computer Science, Carnegie Mellon University

5000 Forbes Ave., Pittsburgh, PA 15213-3891, USA

{sjauhar, yvchen, fmetze}@cs.cmu.edu

## Abstract

This paper presents a graph-based model that integrates prosodic features into an unsupervised speech summarization framework without any lexical information. In particular it builds on previous work using mutually reinforced random walks, in which a two-layer graph structure is used to select the most salient utterances of a conversation. The model consists of one layer of utterance nodes and another layer of prosody nodes. The random walk algorithm propagates scores between layers to use shared information for selecting utterance nodes with highest scores as summaries. A comparative evaluation of our prosody-based model against several baselines on a corpus of academic multi-party meetings reveals that it performs competitively on very short summaries, and better on longer summaries according to ROUGE scores as well as the average relevance of selected utterances.

## 1 Introduction

Automatic extractive speech summarization (Hori and Furui, 2001) has garnered considerable interest in the natural language processing research community for its immediate application in making large volumes of multimedia documents more accessible. Several variants of speech summarization have been studied in a range of target domains, including news (Hori et al., 2002; Maskey and Hirschberg, 2003), lectures (Glass et al., 2007; Chen et al., 2011) and multi-party meetings (Banerjee and Rudnicky, 2008; Liu and Liu, 2010; Chen and Metze, 2012b).

Research in speech summarization – unlike its text-based counterpart – carries intrinsic difficulties, which draw their origins from the noisy nature of the data under consideration: imperfect

ASR transcripts due to recognition errors, lack of proper segmentation, etc. However, it also offers some advantages by making it possible to leverage extra-textual information such as emotion and other speaker states through an incorporation of prosodic knowledge into the summarization model.

A study by Maskey and Hirschberg (2005) on the relevance of various levels of linguistic knowledge (including lexical, prosodic and discourse structure) showed that enhancing a summarizer with prosodic information leads to more accurate and informed results.

In this work we extend the model proposed by Chen and Metze (2012c), where a random walk is performed on a lexico-topical graph structure to yield summaries. They exploited intra- and inter-speaker relationships through partial topic sharing for judging the importance of utterances in the context of multi-party meetings. This paper, on the other hand, enriches the underlying graph structure with prosodic information, rather than lexico-topical knowledge, to model speaker states and emotions.

Also different from Maskey and Hirschberg (2005), we model the multimedia document structure as a graph, which allows for flexibility as well as expressive power in representation. This graph structure provides the easy incorporation of targeted features into the model as well as in-depth analyses of individual feature contributions towards representing speaker information.

To the best of our knowledge this paper presents the first attempt at performing speech summarization using no lexical information in a completely unsupervised setting. Maskey and Hirschberg (2006) use an HMM to perform summarization by relying solely on prosodic features. However, their model – unlike ours – is supervised. The only requirement of the model in this paper is a pre-processing step that segments the audio into “ut-

terances”.

While utterance segmentation may be a non-trivial problem, the possibility of an unsupervised speech summarization model that relies solely on acoustic input is advantageous. Importantly, it does not rely on any training data and circumvents the primary difficulties that plague most speech summarization techniques — namely the noise introduced into the system by imperfect speech recognition.

We evaluate our model on a dataset consisting of multi-party academic meetings (Chen and Metze, 2012b; Banerjee and Rudnicky, 2008). We perform evaluation using the ROUGE metric for automatic summarization, which counts  $n$ -gram overlap between reference and candidate summaries. We also run a post-hoc analysis, which measures the average relevance score of utterances in a candidate summary.

Evaluation results indicate that our model outperforms a number of baselines across varying experimental settings in all but the shortest summaries. We hence claim that our model is a robust, flexible, and effective framework for unsupervised speech summarization.

The rest of the paper is organized as follows. Section 2 describes the prosodic features encoded in the model and how they are extracted. Section 3 presents the construction of the two-layer graph and mutually reinforced random walk for propagating information through the graph. Section 4 shows experimental results of applying the proposed model to the dataset of academic meetings and discusses the effects of prosody on summarization. Section 5 concludes.

## 2 Prosodic Feature Extraction

As previously stated, the only pre-requisite of the model proposed in this paper is a segmentation of the input document into chunks that are dictated by some meaningful notion of utterances. Once the audio has been segmented utterance-wise, the rest of the pipeline is effectively agnostic to all but its acoustic properties.

Given a set of pre-segmented audio files, we extract the following prosodic features from them using PRAAT scripts (Huang et al., 2006).

- Number of syllables and number of pauses.
- Duration time, – which is the speaking time including pauses – and the phonation time, –

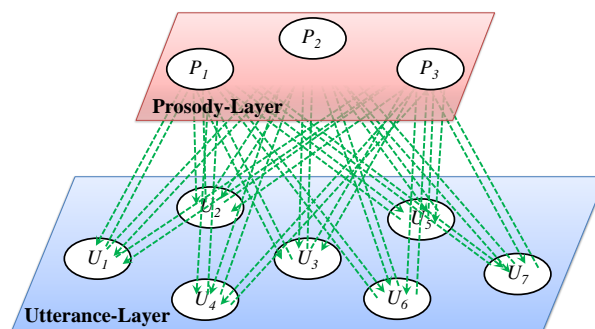


Figure 1: A simplified example of the two-layer graph considered, where a type of prosody  $P_i$  is represented as a node in prosody-layer and an utterance  $U_j$  is represented as a node in utterance-layer of the two-layer graph.

which is the speaking time excluding pauses.

- Speaking rate and articulation rate, which are the number of syllables divided by the duration time and phonation time, respectively.
- The average, maximal and minimal fundamental frequencies measured in Hz (which objectify the perceptive notion of pitch).
- The energy measured in  $\text{Pa}^2/\text{sec}$  and the intensity measured in dB.

The inclusion of the features above into the model was motivated by their possible contribution to the notion of “important utterances” in a dialogue. For example, intuitively, pitch is a vocal channel for emotions, such as anger, or embarrassment. It may thus contribute, via the emotional investment of the speaker to the importance of her utterances. Similarly, the variation of energy over an utterance results in its perceived loudness, thus possibly permitting the inference of emphasis or stress to particular utterances by speakers. Again, speech rate often acts as a latent channel for communication of information, where excitement or emphasis is implicitly conveyed by a speaker.

## 3 Two-Layer Mutually Reinforced Random Walk

In this section we describe our method for modelling speech data as a two-layered interconnected graph structure and run the mutually reinforced random-walk algorithm for summarization.

Given an input speech document that is suitably segmented into utterance chunks, we construct a linked two-layer graph  $G$  containing an utterance set  $V_U$  and a prosody set  $V_P$ . Each node of the graph  $U_i \in V_U$  corresponds to a single utterance as obtained from the pre-processing ‘‘chunking’’ step. Every node  $P_i \in V_P$  illustrates a single prosodic features incorporated into the model.

Figure 1 shows a simplified example of such a two-layered graph.  $G = \langle V_U, V_P, E_{UP}, E_{PU} \rangle$ , where  $V_U = \{U_i\}$ ,  $V_P = \{P_i\}$ ,  $E_{UP} = \{e_{ij} \mid U_i \in V_U, P_j \in V_P\}$ , and  $E_{PU} = \{e_{ij} \mid P_i \in V_P, U_j \in V_U\}$ . Here,  $E_{UP}$  and  $E_{PU}$  represent the sets of directional edges between utterances and prosodic nodes with different directions (Cai and Li, 2012).

Based on these sets of directional edges we further define  $L_{UP} = [w_{i,j}]_{|V_U| \times |V_P|}$  and  $L_{PU} = [w_{j,i}]_{|V_P| \times |V_U|}$ . The matrices  $L_{UP}$  and  $L_{PU}$  effectively encode the directional relationship between utterances and prosodic features. More concretely, for example, the entry  $w_{i,j}$  of  $L_{UP}$  is the value of the prosodic feature  $P_j$  extracted from the utterance  $U_i$ . Row-normalization is performed on  $L_{UP}$  and  $L_{PU}$  (Shi and Malik, 2000). It may be noted that, as a consequence,  $L_{UP}$  is different from  $L_{PU}^T$ .

Traditional random walk only operates on a single layer of the graph structure and integrates the initial similarity scores with the scores propagated from other utterance nodes (Chen et al., 2011; Chen and Metze, 2012a; Hsu et al., 2007). The approach adopted in this paper, however, considers prosodic information by propagating information between layers based on external mutual reinforcement (Chen and Metze, 2012c).

Effectively the working of the algorithm stems from two interrelated intuitions. On the one hand, utterances that evidence more pronounced signs of important prosodic features should themselves be judged as more salient. On the other hand, prosodic features in salient utterances that are recorded with higher values should themselves be deemed as more important.

The advantage of the algorithm is that it is entirely unsupervised and allows for the integration of knowledge-rich target specific features. The mathematical formulation of the algorithm is presented as follows.

Given some initial scores  $F_U^{(0)}$  and  $F_P^{(0)}$  for utterance and prosody nodes respectively, the update

rule is given by:

$$\begin{cases} F_U^{(t+1)} = (1 - \alpha)F_U^{(0)} + \alpha \cdot L_{UP}F_P^{(t)} \\ F_P^{(t+1)} = (1 - \alpha)F_P^{(0)} + \alpha \cdot L_{PU}F_U^{(t)} \end{cases} \quad (1)$$

Here  $F_U^{(t)}$  and  $F_P^{(t)}$  integrate the initial importance associated with their respective nodes with the score obtained by between-layer propagation at a given iteration  $t$ .

Hence, the scores in each layer are mutually updated by the scores from the other layer, iteratively. In particular, utterances that exhibit more pronounced signs of important prosodic feature are progressively scored higher. At the same time, prosodic features that appear with higher values in salient utterances become progressively more important.

For the utterance set, the update rule increments the importance of nodes with the combination  $L_{UP}F_P^{(t)}$ . This latter term can be considered as the score from linked nodes in the prosody set, weighted by prosodic feature values. Finally, an  $\alpha$  value encodes the trade-off between initial utterance weight and information sharing via propagation. The algorithm converges satisfying (2).

$$\begin{cases} F_U^* = (1 - \alpha)F_U^{(0)} + \alpha \cdot L_{UP}F_P^* \\ F_P^* = (1 - \alpha)F_P^{(0)} + \alpha \cdot L_{PU}F_U^* \end{cases} \quad (2)$$

Additionally  $F_U^*$  has an analytical solution which is given by:

$$\begin{aligned} F_U^* &= (1 - \alpha)F_U^{(0)} \\ &+ \alpha \cdot L_{UP} \left( (1 - \alpha)F_P^{(0)} + \alpha \cdot L_{PU}F_U^* \right) \\ &= (1 - \alpha)F_U^{(0)} + \alpha(1 - \alpha)L_{UP}F_P^{(0)} \\ &+ \alpha^2 L_{UP}L_{PU}F_U^* \\ &= \left( (1 - \alpha)F_U^{(0)} e^T + \alpha(1 - \alpha)L_{UP}F_P^{(0)} e^T \right. \\ &+ \left. \alpha^2 L_{UP}L_{PU} \right) F_U^* \\ &= MF_U^*, \end{aligned} \quad (3)$$

where  $e = [1, 1, \dots, 1]^T$ . The closed-form solution  $F_U^*$  of (3) is the dominant eigenvector of  $M$  (Langville and Meyer, 2005).

It may be noted that for the practical implementation of the algorithm, we set the initial scores of utterance nodes  $F_U^{(0)}$  and prosodic nodes  $F_P^{(0)}$  to have equal importance. Also we empirically set  $\alpha = 0.9$  for all our experiments because several studies have shown that  $(1 - 0.9)$  is a proper damping factor (Hsu et al., 2007; Brin and Page, 1998).

## 4 Experiments

### 4.1 Pre-processing – Time Alignment

We have previously stressed that while our model is independent from the lexical representation of an audio document, it does rely on a pre-processing step that chunks the document into individual utterances. It is noted that this may not be a trivial task.

Speaker diarization (Tranter and Reynolds, 2006) and utterance segmentation (Christensen et al., 2005; Geertzen et al., 2007) are open areas of research in the NLP community. Systems developed for these purposes may be used to produce the initial chunking required by our model. In this paper, however, we do not explore these methods and instead rely on segmentation obtained from manually produced textual transcripts. This is to study the efficacy of our model in isolation.

A second reason for using textual transcripts is the presentation of experimental evaluation. This form of data allows for tangible results that are obtained through evaluation metrics such as ROUGE, which rely on measuring  $n$ -gram overlap between reference and candidate summaries. Furthermore, the resulting textual surface form and summaries are more “semantically” interpretable as well.

To associate prosodic information with the textual realization of each utterance in a manual transcript, a preprocessing step requires time alignments between the audio and the corresponding text of each utterance. Note that this step is unnecessary in the case when manual transcripts are not present, and utterance chunking is obtained from some other, automatic means. The time alignment is then implicitly obtained in the process of utterance segmentation.

To accomplish the alignment in our experimental framework, a speech recognizer is first used to produce an ASR output of the audio document. A by-product of this step is that each recognized token contains an inherent time signature. Using Viterbi alignment between the ASR output and manual transcription the time signatures from the audio is projected onto each manually transcribed utterance.

We experimented with Viterbi alignment at a number of different levels of granularity including token level, character level, and phoneme level (via conversion of text to phonetic representation using the CMU pronunciation dictionary (Weide,

1998)). The latter was empirically found to produce the most fine-grained and precise alignments, and was consequently used in all our experiments.

### 4.2 Corpus

The dataset used in our evaluation is the same one previously employed by Chen and Metze (2012b). It consists of 10 meetings held between April and June 2006, with largely overlapping participants and topics of discussion. There were a total of 6 unique participants, with each meeting involving between 2 and 4 individual speakers. SmartNotes (Banerjee and Rudnicky, 2008) was used to record both the audio and the notes for each meeting.

The average duration of a meeting in the dataset was approximately 28 minutes, and the total number of utterances was 7123. We only use the manual transcripts of the meetings to actually evaluate our model, although ASR transcripts were used for time alignment.

The reference summaries are produced by selecting the set of the most “noteworthy” utterances. Two annotators manually labelled the degree of “noteworthiness” (on a relevance scale of 1 to 3) for each utterance. We extract all the utterances with a “noteworthiness” level of 3 to form the reference summary of each meeting.

### 4.3 Baselines

Several baselines were used for comparison against our model and are described below.

1. **Longest:** The first baseline simply selects the longest utterances to form a summary of a document (where the length of the extracted summary is based on the desired ratio). We define the length of utterances by the number of tokens they contain.
2. **Begin:** A second variant of this baseline selects the utterances that appear in the beginning of the document.
3. **LTE:** The third baseline is a summary produced by using Latent Topic Entropy (LTE) (Kong and Lee, 2011). This measure essentially estimates the “focus” of an utterance. Hence, theoretically, a lower topic entropy relates to a more topically informative utterance, which in turn translates into a noteworthy utterance to include in a summary.
4. **TF-IDF** The final baseline uses basic TF-IDF to measure the importance of utterances, by



F-measure		ROUGE-1			ROUGE-L		
		10%	20%	30%	10%	20%	30%
Baseline	Longest	34.05	52.48	61.11	33.66	52.10	60.77
	Begin	<b>35.45</b>	54.42	64.63	<b>35.28</b>	54.18	64.37
	LTE	35.16	54.67	64.97	35.03	54.54	64.76
	TFIDF	32.01	51.33	63.11	31.89	51.08	62.84
This Paper		35.33	<b>55.17</b>	<b>65.60</b>	35.09	<b>54.90</b>	<b>65.36</b>

Table 1: ROUGE scores (%) on multi-party meeting dataset

taking the averaged TF-IDF score over each of its individual words.

It may be noted that the topic distribution of words as well as their IDF scores were obtained by computing statistics over all ten meetings in our experimental dataset.

#### 4.4 Evaluation Metrics

Our automated evaluation utilizes the standard DUC (Document Understanding Conference) evaluation metric, ROUGE (Lin, 2004), which measures recall over various  $n$ -gram statistics between a system-generated summary and a set of summaries produced by humans. F-measures for ROUGE-1 (unigram) and ROUGE-L (longest common subsequence) can be evaluated in exactly the same way.

We also use a post-hoc evaluation metric to measure the average “importance” of utterances in a summary. This metric associates a relevance score to a summary by taking the averaged noteworthiness score of each utterance, as obtained from human annotators.

#### 4.5 Results and Discussion

We ran each of the baseline summarizers as well as the system proposed in this paper to produce 10%, 20% and 30% summaries of each of the meetings in the dataset. The percentage of a summary was determined by selecting the top  $k$  utterances (as determined by a given system) until the desired ratio between the number of tokens in the summary to the total length of its corresponding meeting was met.

Evaluation results on the ROUGE metric are presented in Table 1. They reveal that the performance of our prosody-based model is competitive with the other baselines on the shortest 10% summaries. In fact it ranks second, only scoring lower than the baseline that considers the beginning of a document as a summary. Additionally, on the

longer 20% and 30% summaries, the system outperforms all the baselines.

We believe that in the case of very short summaries, the nature of the data under consideration biases the evaluation of the “begin” baseline. This is because the meetings generally commence with a presentation of an agenda which contains key terms that are likely to be discussed during the course of the rest of the session. In this scenario a metric such as ROUGE – which effectively measures  $n$ -gram overlap – would reward the “begin” summary for including key terms that appear several times in the gold standard summaries.

However for longer summaries, where lexical variation is more pronounced, prosodic information provides a robust source of intelligence to select noteworthy utterances. In fact we are surprised that it outperforms the lexically derived LTE and TF-IDF baselines in all evaluation configurations.

Overall, these results seem to suggest that our model is able to capture latent speaker information and incorporate it effectively into the process of extractive summarization.

We further test this conclusion by conducting a post-hoc analysis, where we examine the average “importance” of utterances in the summary produced by a particular system. More specifically, we measure the average relevance score – ranging on a scale of 1 to 3 – of the utterances, where the score of each utterance is derived from its noteworthiness level as judged by human annotators (Banerjee and Rudnicky, 2008). The results of this analysis are presented in Table 2.

While the “begin” baseline is able to produce summaries with the highest relevance score for the shortest 10% summaries, our model outperforms all other systems on the longer 20% and 30% summaries. Moreover, it is competitive with the “begin” baseline even on the shortest summaries and scores higher than the other baselines. These re-

Avg. Relevance		10%	20%	30%
Baseline	Longest	2.299	2.272	2.283
	Begin	<b>2.464</b>	2.402	2.398
	LTE	2.334	2.369	2.367
	TFIDF	2.355	2.363	2.375
This paper		2.454	<b>2.422</b>	<b>2.411</b>

Table 2: Avg. relevance scores on multi-party meeting dataset

sults align with the findings in Table 1.

As an auxiliary analysis we also extract the converged scores of prosody nodes and rank them in order to analyze their effectiveness. The ranking reveals that the number of pauses in an utterance, its minimum and average pitch, and its intensity tend to be the most predictive features. In the context of academic meetings the number of pauses may be indicative of the time a speaker takes to formulate and articulate his/her thoughts. Thus more pauses may indicate utterances that have been more carefully crafted and therefore include more relevant content. Pitch and intensity are generally good measures of important information, because speakers tend to use them to express emotion. This fact has previously been successfully leveraged for key term extraction (Chen et al., 2010).

Conversely the duration time of the utterance, the number of syllables, and the energy are the least predictive features. With the exception of energy, the other two features can be considered as a surrogate measure for the length of utterances. This parallels what the “longest” utterance baseline performs lexically. The finding corresponds to the results from Tables 1 and 2, which show that this baseline does not produce particularly relevant summaries.

## 5 Conclusion

Our paper proposes a novel approach to integrating speaker-state information, through the incorporation of prosodic knowledge into an unsupervised model for extractive speech summarization. We have also shown the first attempt at performing unsupervised speech summarization without using lexical information.

We have presented experiments on a dataset of academic meetings involving spoken interactions between multiple parties. Evaluation results indicate that our model extracts relevant utterances

as summaries, both from the perspective of automatic evaluation metrics such as ROUGE as well as a post-hoc metric that measured the average relevance score of utterances within summaries. In addition our model compared favorably with a number of heuristic and lexically derived baselines outperforming them in all but one scenario. This substantiates its claim to a robust and viable method for completely unsupervised speech summarization.

## References

- Satanjeev Banerjee and Alexander I. Rudnicky. 2008. An extractive-summarization baseline for the automatic detection of noteworthy utterances in multi-party human-human dialog. In *Proceedings of The 2nd IEEE Workshop on Spoken Language Technology (SLT)*, pages 177–180. IEEE.
- Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117.
- Xiaoyan Cai and Wenjie Li. 2012. Mutually reinforced manifold-ranking based relevance propagation model for query-focused multi-document summarization. *IEEE Transactions on Acoustics, Speech and Language Processing*, 20:1597–1607.
- Yun-Nung Chen and Florian Metze. 2012a. Integrating intra-speaker topic modeling and temporal-based inter-speaker topic modeling in random walk for improved multi-party meeting summarization. In *Proceedings of The 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- Yun-Nung Chen and Florian Metze. 2012b. Intra-speaker topic modeling for improved multi-party meeting summarization with integrated random walk. In *Proceedings of The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 377–381, Montréal, Canada, June. Association for Computational Linguistics.
- Yun-Nung Chen and Florian Metze. 2012c. Two-layer mutually reinforced random walk for improved multi-party meeting summarization. In *Proceedings of The 4th IEEE Workshop on Spoken Language Technology (SLT)*.
- Yun-Nung Chen, Yu Huang, Sheng-Yi Kong, and Lin-Shan Lee. 2010. Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features. In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pages 265–270. IEEE.

- Yun-Nung Chen, Yu Huang, Ching-Feng Yeh, and Lin-Shan Lee. 2011. Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms. In *Proceedings of The 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- Heidi Christensen, BalaKrishna Kolluru, Yoshihiko Gotoh, and Steve Renals. 2005. Maximum entropy segmentation of broadcast news. *IEEE Signal Processing Society Press*.
- Jeroen Geertzen, Volha Petukhova, and Harry Bunt. 2007. A multidimensional approach to utterance segmentation and dialogue act classification. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp*, pages 140–149.
- James R Glass, Timothy J Hazen, D Scott Cyphers, Igor Malioutov, David Huynh, and Regina Barzilay. 2007. Recent progress in the mit spoken lecture processing project. In *Proceedings of The 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2553–2556.
- Chiori Hori and Sadaoki Furui. 2001. Advances in automatic speech summarization. *RDM*, 80:100.
- Chiori Hori, Sadaoki Furui, Rob Malkin, Hua Yu, and Alex Waibel. 2002. Automatic speech summarization applied to english broadcast news speech. In *Proceedings of ICASSP*, volume 1, pages 9–12.
- Winston H Hsu, Lyndon S Kennedy, and Shih-Fu Chang. 2007. Video search reranking through random walk over document-level context graph. In *Proceedings of the 15th international conference on Multimedia*, pages 971–980. ACM.
- Zhongqiang Huang, Lei Chen, and Mary Harper. 2006. An open source prosodic feature extraction tool. In *Proceedings of the Language Resources and Evaluation Conference*.
- Sheng-Yi Kong and Lin-Shan Lee. 2011. Semantic analysis and organization of spoken documents based on parameters derived from latent topics. *IEEE Trans. on Audio, Speech and Language Processing*.
- Amy N Langville and Carl D Meyer. 2005. A survey of eigenvector methods for web information retrieval. *SIAM Review*, 47(1):135–161.
- 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.
- Fei Liu and Yang Liu. 2010. Using spoken utterance compression for meeting summarization: A pilot study. In *Proceedings of The 3rd IEEE Workshop on Spoken Language Technology (SLT)*.
- Sameer Raj Maskey and Julia Hirschberg. 2003. Automatic summarization of broadcast news using structural features. In *Proceedings of Eurospeech*.
- Sameer Maskey and Julia Hirschberg. 2005. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Proceedings of InterSpeech*.
- Sameer Maskey and Julia Hirschberg. 2006. Summarizing speech without text using hidden markov models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 89–92. Association for Computational Linguistics.
- Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- Sue E Tranter and Douglas A Reynolds. 2006. An overview of automatic speaker diarization systems. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1557–1565.
- RL Weide. 1998. The cmu pronunciation dictionary, release 0.6.

# Mining the Gaps: Towards Polynomial Summarization

**Marina Litvak**

Sami Shamoon College of Engineering  
Beer Sheva, Israel  
marinal@sce.ac.il

**Natalia Vanetik**

Sami Shamoon College of Engineering  
Beer Sheva, Israel  
natalyav@sce.ac.il

## Abstract

The problem of text summarization for a collection of documents is defined as the problem of selecting a small subset of sentences so that the contents and meaning of the original document set are preserved in the best possible way. In this paper we present a linear model for the problem of text summarization<sup>1</sup>, where a summary preserves the information coverage as much as possible in comparison to the original document set. We reduce the problem of finding the best summary to the problem of finding the point on a convex polytope closest to the given hyperplane, and solve it efficiently with the help of fractional (polynomial-time) linear programming. The experimental results show the superiority of our approach over most of the systems participating in the generic multi-document summarization task (MultiLing) of the TAC 2011 competition.

## 1 Introduction

Automated text summarization is an active field of research in various communities like Information Retrieval (IR), Natural Language Processing (NLP), and Text Mining (TM).

Some authors reduce summarization to the maximum coverage problem (Takamura and Okumura, 2009; Gillick and Favre, 2009) that, despite a great performance, is known as NP-hard (Khuller et al., 1999). Linear Programming helps to find an accurate approximated solution to this problem and became very popular in summarization field in the last years (Gillick and Favre, 2009; Woodsend and Lapata, 2010; Hitoshi Nishikawa and Kikui, 2010; Makino et al.,

<sup>1</sup>This work was partially funded by U.S. Department of Navy, Office of Naval Research.

2011). However, most mentioned works use exponential number of constraints or Integer Linear Programming which is an NP-hard problem.

Trying to solve a trade-off between summary quality and time complexity, we propose a novel summarization model solving the approximated maximum coverage problem by linear programming in polynomial time. We measure information coverage by terms<sup>2</sup> and strive to obtain a summary that preserves the optimal value of the chosen objective function as much as possible in comparison to the original document. Various objective functions combining different parameters like term's position and its frequency are introduced and evaluated.

Our method ranks and extracts significant sentences into a summary and it can be generalized for both single-document and multi-document summarization. Also, it can be easily adapted to cross-lingual/multilingual summarization.

Formally speaking, in this paper we introduce (1) a novel text representation model expanding a classic Vector Space Model (Salton et al., 1975) to Hyperplane and Half-spaces, (2) re-formulated extractive summarization problem as an optimization task and (3) its solution using linear or quadratic programming. The main challenge of this paper is a new text representation model making possible to represent an exponential number of extracts without computing them explicitly, and finding the optimal one by simple minimizing a distance function in polynomial time.

## 2 Our Method

### 2.1 Definitions

We are given a set of sentences  $S_1, \dots, S_n$  derived from a document or a cluster of related documents. Meaningful words in these sentences are entirely described by terms  $T_1, \dots, T_m$ . Our goal is to find a

<sup>2</sup>normalized meaningful words

subset  $S_{i_1}, \dots, S_{i_k}$  consisting of sentences such that (1) there are at most  $N$  terms in these sentences, (2) term frequency is preserved as much as possible w.r.t. the original sentence set, (3) redundant information among  $k$  selected sentences is minimized.

We use the standard sentence-term matrix,  $A = (a_{ij})$  of size  $m \times n$ , for initial data representation, where  $a_{ij} = k$  if term  $T_i$  appears in the sentence  $S_j$  precisely  $k$  times. Here, columns of  $A$  describe sentences and rows describe terms. Since we are not interested in redundant sentences, in the case of multi-document summarization, we can initially select meaningful sentences by clustering all the columns as vectors in  $\mathbb{R}^n$  and choose a single representative from each cluster. In this case columns of  $A$  describe representatives of sentence clusters. The total number of words (term appearances) in the document, denoted by  $S$ , can be computed from the matrix  $A$  as

$$S = \sum_i \sum_j a_{ij} \quad (1)$$

**Example 1.** Given the following text of  $n = 3$  sentences and  $m = 5$  (normalized) terms:

$S_1 = A \text{ fat cat is a cat that eats fat meat.}$   
 $S_2 = My \text{ cat eats fish but he is a fat cat.}$   
 $S_3 = All \text{ fat cats eat fish and meat.}$

Matrix  $A$  corresponding to the text above has the following shape:

$$\begin{array}{l} T_1 = \text{"fat"} \\ T_2 = \text{"cat"} \\ T_3 = \text{"eat"} \\ T_4 = \text{"fish"} \\ T_5 = \text{"meat"} \end{array} \begin{bmatrix} S_1 & S_2 & S_3 \\ a_{11} = 2 & a_{12} = 1 & a_{13} = 1 \\ a_{21} = 2 & a_{22} = 2 & a_{23} = 1 \\ a_{31} = 1 & a_{32} = 1 & a_{33} = 1 \\ a_{41} = 0 & a_{42} = 1 & a_{43} = 1 \\ a_{51} = 1 & a_{52} = 0 & a_{53} = 1 \end{bmatrix}$$

where  $a_{ij}$  are term counts. The total count of terms in this matrix is

$$S = \sum_{i=1}^5 \sum_{j=1}^3 a_{ij} = 16$$

Our goal is to find subset  $i_1, \dots, i_k$  of  $A$ 's columns so that the chosen submatrix represents the best possible summary under some constraints. Since it is hard to determine what is the best summary mathematically (this task is usually left to human experts), we wish to express summary quality as a linear function of the underlying matrix. We strive to find a summary that gives an optimal value once the function in question has been determined.

## 2.2 Text Preprocessing

In order to build the matrix and then the polytope model, one needs to perform the basic text preprocessing including sentence splitting and tokenization. Also, additional steps like stopwords removal, stemming, synonym resolution, etc. may be performed for resource-rich languages. Since the main purpose of these methods is to reduce the matrix dimensionality, the resulted model will be more efficient.

## 2.3 Polytope as a document representation

We represent every sentence by a hyperplane, and all sentences derived from a document form a hyperplane intersections (polytope). Then, all possible extracts can be represented by subplanes of our hyperplane intersections and as such that are not located far from the boundary of the polytope. Intuitively, the boundary of the resulting polytope is a good approximation for extracts that can be generated from the given document. We view every column of the sentence-term matrix as a *linear constraint* representing a hyperplane in  $\mathbb{R}^m$ . An occurrence of term  $t_i$  in sentence  $S_j$  is represented by variable  $x_{ij}$ . The maximality constraint on the number of terms in the summary can be easily expressed as a constraint on the sum of these variables.

**Example 2.** This example demonstrates variables corresponding to the  $5 \times 3$  matrix  $A$  of Example 1.

$$\begin{array}{l} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{array} \begin{bmatrix} S_1 & S_2 & S_3 \\ x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{bmatrix}$$

Every sentence in our document is a hyperplane in  $\mathbb{R}^m$ , defined with columns of  $A$  and variables representing terms in sentences:

$$\begin{aligned} A[[j]] &= [a_{1j}, \dots, a_{mj}] \\ \mathbf{x}_j &= [x_{1j}, \dots, x_{mj}] \text{ for all } 1 \leq j \leq n \end{aligned}$$

We define a system of linear inequalities

$$\begin{aligned} A[[j]] \cdot \mathbf{x}_j^T &= \sum_{i=1}^m a_{ij} x_{ij} \leq \\ &\leq A[[j]] \cdot \mathbf{1}^T = \sum_{i=1}^m a_{ij} \end{aligned} \quad (2)$$

Every inequality of this form defines a hyperplane  $H_i$  and its lower half-space specified by equation (2):

$$A[[j]] \cdot \mathbf{x}_j^T = A[[j]] \cdot \mathbf{1}^T$$

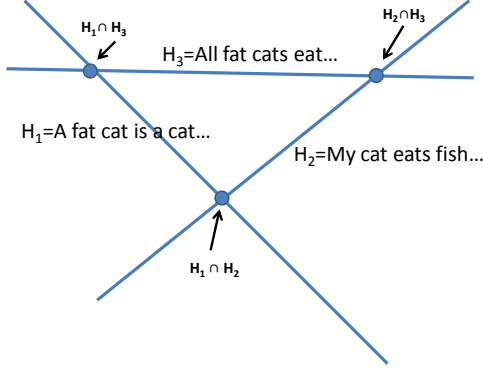


Figure 1: Two-dimensional projection of hyperplane intersection.

and with normal vector  $\mathbf{n} = (\mathbf{n}_{xy})$

$$\mathbf{n}_{xy} = \begin{cases} a_{xy} & 1 \leq x \leq m \wedge y = j \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

To say that every term is either present or absent from the chosen extract, we add constraints  $0 \leq x_{ij} \leq 1$ . Intuitively, entire hyperplane  $H_i$  and therefore every point  $p \in H_i$  represents sentence  $S_i$ . Then a subset of  $r$  sentences is represented by intersection of  $r$  hyperplanes.

**Example 3.** Sentence-term matrix  $A$  of Example 1 defines the following hyperplane equations.

$$H_1: 2x_{11} + 2x_{21} + x_{31} + x_{51} = 2 + 2 + 1 + 1 = 6$$

$$H_2: x_{12} + 2x_{22} + x_{32} + x_{42} = 5$$

$$H_3: x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 5$$

Here, a summary consisting of the first and the second sentence is expressed by the intersection of hyperplanes  $H_1$  and  $H_2$ . Figure 1 shows how a two-dimensional projection of hyperplanes  $H_1, H_2, H_3$  and their intersections look like.

## 2.4 Summary constraints

We express summarization constraints in the form of linear inequalities in  $\mathbb{R}^{mn}$ , using the columns of the sentence-term matrix  $A$  as linear constraints. Maximality constraint on the number of terms in the summary can be easily expressed as a constraint on the sum of term variables  $x_{ij}$ .

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq T_{\max} \quad (4)$$

**Example 4.** Equation (4) for Example 1,  $T_{\max} = 11$  has the form

$$\begin{aligned} 0 \leq x_{ij} \leq 1, \forall i, j \\ \sum_{i=1}^5 \sum_{j=1}^3 x_{ij} \leq 11 \end{aligned}$$

Additionally, we may have constraints on the maximal  $W_{\max}$  number of words in the summary. We take into account only words that remain in the text after stop-word removal and stemming. The difference between the number terms and the number of words in a summary is that a single term can appear more than once in a sentence. Therefore, the total number of words in the text is expressed by summing up the elements of its term-count matrix. Therefore, maximality constraints for words are expressed by the following linear inequality.

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \leq W_{\max} \quad (5)$$

**Example 5.** Equation (5) for the sentence-term matrix of Example 1 for  $W_{\max} = 11$  has the form

$$\begin{aligned} 2x_{11} + 2x_{21} + x_{31} + x_{51} + \\ + x_{12} + x_{22} + 2x_{32} + x_{42} + \\ + x_{13} + x_{23} + x_{33} + x_{43} + x_{53} \leq 11 \end{aligned}$$

## 2.5 The polytope model

Having defined linear inequalities that describe each sentence in a document separately and the total number of terms in sentence subset, we can now look at them together as a system:

$$\begin{cases} \sum_{i=1}^m a_{i1} x_{i1} \leq \sum_{i=1}^m a_{i1} \\ \dots \\ \sum_{i=1}^m a_{in} x_{in} \leq \sum_{i=1}^m a_{in} \\ \sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq T_{\max} \\ \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \leq W_{\max} \\ 0 \leq x_{ij} \leq 1 \end{cases} \quad (6)$$

First  $n$  inequalities describe sentences  $S_1, \dots, S_n$ , the next two inequalities describes constraints on the total number of terms and words in a summary, and the final constraint determines upper and lower boundaries for all sentence-term variables. Since every inequality in the system (6) is linear, the entire system describes a convex polyhedron in  $\mathbb{R}^{mn}$ , which we denote by  $\mathbf{P}$ . Faces of  $\mathbf{P}$  are determined by intersections of hyperplanes defined in (6).

## 2.6 Objectives and summary extraction

We assume here that the surface of the polyhedron  $\mathbf{P}$  is a suitable representation of all the possible sentence subsets (its size, of course, is not polynomial in  $m$  and  $n$  since the number of vertices of  $\mathbf{P}$  can reach  $O(2^n)$ ). Fortunately, we do not need to scan the whole set of  $\mathbf{P}$ 's surfaces but rather to find

Function	Formula	Description
Maximal Weighted Term Sum ( <i>OBJ</i> <sub>1</sub> )	$\max \sum_{i=1}^m w_i t_i,$ $t_i = \sum_{j=1}^n x_{ji}$	Maximizes the information coverage as a weighted term sum. We used the following types of term weights $w_i$ . (1) POS_EQ, where $w_i = 1$ for all $i$ ; (2) POS_F, where $w_i = \frac{1}{app(i)}$ and $app(i)$ is the index of a sentence in the document where the term $T_i$ first appeared; (3) POS_B, where $w_i = \max\{\frac{1}{app(i)}, \frac{1}{n-app(i)+1}\}$ ; (4) TF, where $w_i = tf(i)$ and $tf(i)$ is the term frequency of term $T_i$ ; (5) TFISF, where $w_i = tf(i) * isf(i)$ and $isf(i)$ is the inverse sentence frequency of $T_i$
Distance Function ( <i>OBJ</i> <sub>2</sub> )	$\min \sum_{i=1}^m (\hat{t}_i - p_i)^2,$ (1) $\hat{t}_i = t_i = \sum_{j=1}^n x_{ji}$ and $\forall i p_i = 1$ , or (2) $\hat{t}_i = \frac{t_i}{\sum_{j=1}^n t_j}$ and $p_i = tf(i)$	Minimizes the Euclidian distance between terms $t = (t_1, \dots, t_m)$ (a point on the polytope $\mathbf{P}$ representing a generated summary) and the vector $p = (p_1, \dots, p_m)$ (expressing document properties we wish to preserve and representing the "ideal" summary). We used the following options for $t$ and $p$ representation. (1) MTC, where $t$ is a summary term count vector and $p$ contains all the terms precisely once, thus minimizing repetition but increasing terms coverage. (2) MTF, where $t$ contains term frequency of terms in a summary and $p$ contains term frequency for terms in documents.
Sentence Overlap ( <i>OBJ</i> <sub>3</sub> )	$\min \sum_{j=1}^n \sum_{k=j+1}^n ovl_{jk},$ $ovl_{jk} = \frac{ S_j \cap S_k }{ S_j \cup S_k } = \frac{\sum_{i=1}^m w(a_{ij}, a_{ik})(x_{ij} + x_{ik})}{\sum_{i=1}^m (a_{ij} + a_{ik})}$	Minimizes the Jaccard similarity between sentences in a summary (denoted by $ovl_{jk}$ for $S_j$ and $S_k$ ). $w(a_{ij}, a_{ik})$ is 1 if the term $T_i$ is present in both sentences $S_j$ and $S_k$ and is 0 otherwise.
Maximal Bigram Sum ( <i>OBJ</i> <sub>4</sub> )	$\max \sum_{i,j} bi_{ij},$ where $\forall i, j, 0 \leq bi_{ij} \leq 1$	Maximizes the information coverage as a bigram sum. Variable $bi_{ij}$ is defined for every bigram $(T_i, T_j)$ in the text.

Table 1: Objective functions for summarization using polytope model.

the point on  $\mathbf{P}$  that optimizes the chosen objective function. Table 1 contains four different objective functions<sup>3</sup> that we used for summarization, along with descriptions of the changes in the model that were required for each function.

Since the LP method not only finds the minimal distance but also presents an evidence to that minimality in the form of a point  $x = (x_{ij})$ , we use the point's data to find what sentences belong to the chosen summary. We check which equations of  $H_i$  the point  $x$  satisfies as equalities. If an equality holds,  $x$  lies on  $H_i$  and therefore the sentence  $S_i$  is contained in the summary. This test is straightforward and takes  $O(mn)$  time. In a case of insufficient summary length, the sentences nearest to the point  $x$  are extracted to a summary in a greedy manner.

### 3 Experiments

In order to evaluate the quality of our approach, we compared our approach to multiple summarizers participated in the generic multi-document summarization task of the TAC 2011 competition (Giannakopoulos et al., 2011) and human

<sup>3</sup>Since our approach is unsupervised, there is no possibility and meaning to use ROUGE, that needs Gold Standard, as an objective.

performance as well. Our software was implemented in Java using Ipsolve (Berkelaar, 1999)<sup>4</sup>. We used the following objective functions, described in Table 1.

- (1) Maximal weighted term sum  $OBJ_1^{weight\_type}$ , where  $weight\_type$  is one of POS\_EQ, POS\_F, POS\_B, TF, TFISF;
- (2) Minimal distance  $OBJ_2^{vector\_type}$ , where  $vector\_type$  is either MTC (Maximal Term Coverage) or MTF (Maximal Term Frequency);
- (3) Minimal sentence overlap  $OBJ_3$ ;
- (4) Maximal bigram sum  $OBJ_4$ .

We conducted the experiments on the MultiLing 2011 (Giannakopoulos et al., 2011) English dataset. MultiLing dataset consists of 10 document sets, 10 documents each one, in seven languages. The original news articles in English were taken from WikiNews<sup>5</sup>, organized into 10 sets, and then summarized. According to the MultiLing summarization task, all systems must generate summaries in size of 250 words at most. Eight systems (ID1-ID8) participated in the pilot and compared to the global baseline (ID9) and the global topline (ID10) systems. Systems A,B and C de-

<sup>4</sup>The software is available upon request.

<sup>5</sup><http://en.wikinews.org/wiki/>

note summaries manually created by human experts. The choice of this dataset is argued by future plans to adapt and evaluate the introduced system to multiple languages.

The automatic summarization evaluation package, ROUGE (Lin, 2004), is used to evaluate the effectiveness of our approach vs. 10 summarizers participated in the MultiLing pilot of the TAC 2011 competition. For fair comparison, only first 250 words<sup>6</sup> were considered in ROUGE statistics. The recall scores of ROUGE- $N$  for  $N \in \{1, 2, 3, 4\}$ , ROUGE- $W$ -1.2, and ROUGE- $SU4$  which are based on  $N$ -gram, Weighted Longest Common Subsequence (WLCS), and Skip-bigram plus unigram, with maximum skip-distance of 4, matching between system summaries and reference summaries, respectively, are reported in Table 2 below.

### 3.1 Experimental Results

As it can be seen from Table 2, our model using unweighted term sum ( $OBJ_1^{POS.EQ}$ ) as an objective function outperforms most of the systems – 6 systems in terms of ROUGE-1, ROUGE-2, ROUGE- $SU4$  and ROUGE- $W$ -1.2, and 8 systems in terms of ROUGE-3 and ROUGE-4. Conversely to our expectations, adding any type of weights to  $OBJ_1$  reduces its performance. Minimizing repetition while increasing terms coverage ( $OBJ_2^{MTC}$ ) shares the same rank with  $OBJ_1^{POS.EQ}$  for most ROUGE metrics. Minimizing distance to a document term frequency vector ( $OBJ_2^{MTF}$ ) performs worse – it outperforms 3, 4, 5, 6, 5 and 3 systems in terms of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE- $SU4$  and ROUGE- $W$ -1.2, respectively. Sentence overlap ( $OBJ_3$ ) and maximal bigram sum ( $OBJ_4$ ) have very close scores, outperforming 3, 5, 5, 6, 5 and 3 systems in terms of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE- $SU4$  and ROUGE- $W$ -1.2, respectively. Generally, optimizing the most of introduced functions generates the near-quality summaries. All functions perform better than the baseline (ID9) system.<sup>7</sup>

## 4 Conclusions and Future Work

In this paper we present a linear programming model for the problem of extractive summariza-

<sup>6</sup>ROUGE.pl -a -x -2 4 -u -c 95 -e data -r 1000 -n 4 -f A -p 0.5 -t 0 -d -l 250

<sup>7</sup>We did not perform tests of statistical significance due to too many comparisons (10 systems vs. 10 objective functions), leaving it as a future work.

tion. We represent the document as a set of intersecting hyperplanes. Every possible summary of a document is represented as an intersection of two or more hyperplanes. We consider the summary to be the best if the optimal value of objective function is preserved during summarization, and translate the summarization problem into a problem of finding a point on a convex polytope which is the closest to the hyperplane describing the "ideal" summary. We introduce multiple objective functions describing the distance between a summary (a point on a convex polytope) and the best summary (the hyperplane).

Since linear programming problem can be solved in polynomial time (see (Karmarkar, 1984), (Khachiyan, 1996; Khachiyan and Todd, 1993)), the time complexity of our approach is polynomial (quadratic, being more precise).

The results of experiments show that our method outperforms most of the systems participated in the MultiLing pilot in terms of various ROUGE metrics. In future, we intend to (1) improve the system's performance by introducing more objective functions and their combinations, (2) adapt our system to multiple languages, and (3) extend our model to query-based summarization.

## Acknowledgments

Authors thank Igor Vinokur for implementing the introduced approach and performing experiments.

## REFERENCES

- Berkelaar, M. (1999). lp-solve free software. <http://lpsolve.sourceforge.net/5.5/>.
- Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., and Varma, V. (2011). TAC 2011 MultiLing Pilot Overview. In *TAC 2011: Proceedings of Text Analysis Conference*.
- Gillick, D. and Favre, B. (2009). A Scalable Global Model for Summarization. In *Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18.
- Hitoshi Nishikawa, Takaaki Hasegawa, Y. M. and Kikui, G. (2010). Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering. In *Coling 2010: Poster Volume*, pages 910–918.



system	rouge-1	system	rouge-2	system	rouge-3	system	rouge-4	system	rouge-SU4	system	rouge-W-1.2
A	0.6690	A	0.4725	A	0.4169	A	0.3951	A	0.4938	A	0.2205
C	0.6519	C	0.4578	C	0.4050	C	0.3851	C	0.4812	C	0.2103
B	0.6457	B	0.4388	B	0.3797	B	0.3589	B	0.4636	B	0.2102
ID10	0.5269	ID10	0.2560	ID10	0.1743	ID10	0.1349	ID10	0.2754	ID10	0.1458
ID2	0.4641	ID2	0.1715	ID3	0.0932	ID3	0.0639	ID2	0.2024	ID2	0.1283
ID4	0.4436	ID3	0.1655	$OBJ_1^{POS.EQ}$	0.0868	$OBJ_1^{POS.EQ}$	0.0636	ID3	0.1941	ID4	0.1227
ID3	0.4266	ID4	0.1507	ID2	0.0849	ID2	0.0551	ID4	0.1900	ID3	0.1199
$OBJ_1^{POS.EQ}$	0.4166	$OBJ_1^{POS.EQ}$	0.1463	ID4	0.0784	$OBJ_1^{POS.B}$	0.0546	$OBJ_1^{POS.EQ}$	0.1819	$OBJ_1^{POS.EQ}$	0.1140
$OBJ_2^{MTC}$	0.4143	$OBJ_2^{MTC}$	0.1426	$OBJ_2^{MTC}$	0.0776	$OBJ_2^{MTC}$	0.0545	$OBJ_2^{MTC}$	0.1777	$OBJ_2^{MTC}$	0.1136
ID5	0.4068	ID5	0.1343	$OBJ_1^{POS.B}$	0.0744	ID4	0.0518	ID5	0.1720	ID1	0.1113
ID1	0.4029	$OBJ_1^{POS.B}$	0.1293	$OBJ_1^{TFISF}$	0.0660	$OBJ_1^{MTF}$	0.0437	$OBJ_1^{POS.B}$	0.1652	ID5	0.1111
$OBJ_1^{TFISF}$	0.3959	$OBJ_1^{TFISF}$	0.1266	ID5	0.0655	$OBJ_1^{TF}$	0.0437	$OBJ_1^{TFISF}$	0.1624	ID8	0.1098
$OBJ_1^{POS.B}$	0.3932	$OBJ_4$	0.1238	$OBJ_3$	0.0649	$OBJ_1^{TFISF}$	0.0436	$OBJ_3$	0.1602	$OBJ_1^{POS.B}$	0.1089
ID7	0.3911	$OBJ_3$	0.1236	$OBJ_4$	0.0649	$OBJ_3$	0.0435	$OBJ_4$	0.1600	$OBJ_3$	0.1079
$OBJ_3$	0.3907	ID8	0.1230	$OBJ_3^{MTF}$	0.0646	$OBJ_4$	0.0435	$OBJ_3^{MTF}$	0.1589	$OBJ_1^{TFISF}$	0.1075
$OBJ_4$	0.3894	$OBJ_2^{MTF}$	0.1222	$OBJ_1^{TF}$	0.0646	$OBJ_1^{POS.F}$	0.0433	$OBJ_1^{TF}$	0.1589	$OBJ_4$	0.1072
$OBJ_2^{MTF}$	0.3874	$OBJ_1^{TF}$	0.1222	$OBJ_1^{POS.F}$	0.0641	ID5	0.0429	ID1	0.1588	$OBJ_2^{MTF}$	0.1067
$OBJ_1^{TF}$	0.3874	$OBJ_1^{POS.F}$	0.1216	ID8	0.0613	ID8	0.0409	$OBJ_1^{POS.F}$	0.1587	$OBJ_1^{POS.F}$	0.1063
$OBJ_1^{POS.F}$	0.3872	ID1	0.1191	ID1	0.0552	ID1	0.0367	ID8	0.1579	ID7	0.1038
ID8	0.3860	ID9	0.1053	ID6	0.0489	ID6	0.0331	ID9	0.1444	$OBJ_1^{TF}$	0.1000
ID9	0.3737	ID6	0.1043	ID9	0.0455	ID9	0.0277	ID6	0.1434	ID9	0.0983
ID6	0.3543	ID7	0.0923	ID7	0.0301	ID7	0.0165	ID7	0.1423	ID6	0.0948

Table 2: Evaluation results. MultiLing 2011. English.

- Karmarkar, N. (1984). New polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395.
- Khachiyan, L. G. (1996). Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21:307–320.
- Khachiyan, L. G. and Todd, M. J. (1993). On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61:137–159.
- Khuller, S., Moss, A., and Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26.
- Makino, T., Takamura, H., and Okumura, M. (2011). Balanced coverage of aspects for text summarization. In *TAC '11: Proceedings of Text Analysis Conference*.
- Salton, G., Yang, C., and Wong, A. (1975). A vector-space model for information retrieval. *Communications of the ACM*, 18.
- Takamura, H. and Okumura, M. (2009). Text summarization model based on maximum coverage problem and its variant. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 781–789.
- Woodsend, K. and Lapata, M. (2010). Automatic Generation of Story Highlights. In *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574.

# Detecting Domain Dedicated Polar Words

Raksha Sharma, Pushpak Bhattacharyya

Dept. of Computer Science and Engineering

IIT Bombay, Mumbai, India

{raksha,pb}@cse.iitb.ac.in

## Abstract

There are many examples in which a word changes its polarity from domain to domain. For example, *unpredictable* is positive in the movie domain, but negative in the product domain. Such words cannot be entered in a “universal sentiment lexicon” which is supposed to be a repository of words with polarity invariant across domains. Rather, we need to maintain separate domain specific sentiment lexicons. The main contribution of this paper is to present an effective method of generating a *domain specific sentiment lexicon*. For a word whose domain specific polarity needs to be determined, the approach uses the Chi-Square test to detect if the difference is *significant* between the counts of the word in positive and negative polarity documents. We extract 274 words that are polar in the movie domain, but are not present in the universal sentiment lexicon. Our overall accuracy is around 60% in detecting movie domain specific polar words.

## 1 Introduction

Sentiment analysis (SA) has attracted a great deal of attention in recent times (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000; Pang et al., 2002; Turney, 2002; Yu and Hatzivassiloglou, 2003; Hu and Liu, 2004; Esuli and Sebastiani, 2005; Breck et al., 2007). The SA task is to predict the sentiment orientation of a text (document/para/sentence) by analyzing the polarity of words present in the text. A lexicon of sentiment bearing words is of great help in such tasks.

Sentiment lexicons are of two types: universal and domain specific. Words like ‘good’ and ‘bad’ have uniform polarity across all domains,

and so are members of universal sentiment lexicon. A word like ‘unpredictable’, on the other hand, is positive in the movie domain (‘unpredictable plot’), but negative in the car domain (‘unpredictable steering’). Such a word should be entered as positive in the sentiment lexicon of the movie domain and as negative in the sentiment lexicon of the car domain.

There are many “universal sentiment lexicons” like SentiWordNet<sup>1</sup>, subjectivity lexicon<sup>2</sup> by Wiebe, list of positive and negative opinion words<sup>3</sup> by Liu. These lexica contain only those polar words which have the same polarity in all domains. In this paper, we use the universal sentiment lexicon published by Wiebe.

Using resources like Wikipedia and SentiWordNet to determine polarity of a domain specific word may lead to wrong sentiment detection. The motivation for our work comes from addressing this problem. We would like to create domain specific sentiment lexicons.

Our technique for detecting domain specific polar words is inspired by the work done by Cheng and Zhulyn (2012). They used the Pearson’s Chi-Square test to find the top 200 words most indicative of positive sentiment and the top 200 words most indicative of negative sentiment from the corpus itself. They used these words as the lexicon for the hitting 2-gram language model. They observed that the hitting 2-gram model achieves far greater accuracy than other language models. In their work, they used the categorical Chi-Square test to determine the score of a word with positive sense and negative sense. Their Chi-Square test gives weightage also to those documents in which the word is absent while calculating the score. However, their idea of selecting hitting words, considers multiple occurrences of a word in a single doc-

<sup>1</sup><http://sentiwordnet.isti.cnr.it/>

<sup>2</sup><http://mpqa.cs.pitt.edu/>

<sup>3</sup><http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html>

ument as one. This leads to loss of information that can help in deciding the correct polarity of a word from the corpus. We use the *goodness of fit Chi Square* test, that takes into account the total occurrences of a word in the corpus to assign the score. This test allows us to compare a collection of categorical data with some theoretical expected distribution<sup>4</sup>.

Our proposed method identifies sentiment words from the corpus. Wiebi (2000) observes that the probability of a sentence being subjective given that there is at least one adjective in the sentence is 55.8%. So we presently focus on adjectives. The key idea is that if a word can have both positive or negative polarity, then it should be uniformly distributed between positive and negative files. For this purpose, we take an equal number of positive and negative reviews from the same domain. So, the expected count of the word in positive and negative reviews is half of the total count in the corpus. This is the null hypothesis.

If the word satisfies the Chi-Square test, it indicates that there is a significant difference between the expected and observed count of the word. Hence, the null hypothesis should be rejected and it should be considered that this deviation from expected value is not by chance, but because of the domain specific polarity of the word, which makes the word more frequent in one of positive or negative reviews.

The road map for the rest of the paper is as follows: Section 2 describes the previous work done in the direction of sentiment lexicon. Section 3 elaborates on the generation of domain specific polar words through the Chi-Square test. Section 4 gives the experimental set up. In section 5, we present results along with discussions. We conclude the paper with points for future work in section 6.

## 2 Related Work

Extensive work has been done in the area of universal sentiment lexicon using corpora based approaches. Wiebe (2000) focused on the problem of identifying subjective adjectives with the help of the corpus. They proposed an approach to find subjective adjectives using the results of a method for clustering words according to their distributional similarity, seeded by a small number of simple adjectives. These adjectives were extracted

from a manually annotated corpus. The basic idea is that subjective words are similar in distribution as they share pragmatic usages. However, the approach is unable to predict sentiment orientations of the found subjective adjectives.

Some evidence exists in the area of domain specific sentiment lexicon. The work of Kanayama and Nasukawa [2006], demonstrates the extraction of domain specific sentiment words in Japanese text. They exploited clause level context coherency to find candidate words for domain specific sentiment lexicon from sentences that appear successively with sentences containing a word from the seed set. The seed set is the set of strong universally polar words. The intuition is that sentences appearing in contexts tend to have the same polarities, so if one of them contains sentiment words, the other successive sentences are expected to contain sentiment words too, with the same polarity. Then, they use a statistical estimation based method to determine whether the candidates are appropriate sentiment words. However, the idea of using a seed set to extract purely domain dependent words may lead to wrong polarity.

Qiu et al. (2009) exploited the relationship between sentiment words and product features that the sentiment words modify in a domain dependent corpus. They used sentiment words and product features to extract new sentiment words. The extraction rules are designed, based on relations described in dependency trees. Their method also begins with a seed set. They proposed that a feature should receive the same polarity in a review and the words extracted by this feature will receive polarity of feature. However, the reviewer may associate polarity with a feature of time. If time changes, his views for a feature may change in the same review. To understand this fact consider the following example.

*“When I purchased this camera, the battery was good, but now it is disastrous”.*

Qui et al. (2009) considered ‘camera’, ‘DVD player’, and ‘MP3 player’ as one domain. However, *grainy* and *blurred* are negative in the camera domain, but neutral for ‘DVD’ and ‘MP3 player’. Our work is independent of features and the seed list. It only needs a sufficient equal number of positive and negative review files written by a reliable source.

<sup>4</sup><http://math.hws.edu/javamath/ryan/ChiSquare.html>

### 3 The Proposed Method

In this paper, we focus on finding sentiment words for the movie domain with their polarity as positive or negative. Finding movie domain specific polar words is an appealing task for several reasons. First, providing polarity information about movie reviews is a useful service. Its proof is the popularity of several film review websites<sup>5</sup>. Second, movie reviews are harder to classify than reviews of other products (Turney, 2002) and so is the classification of sentiment words. Our data contains 1000 positive and 1000 negative reviews, all written before 2004<sup>6</sup>. Movie reviews are accompanied by plot descriptions and plot is not a part of the reviewer's opinion of the movie. So presence of polar words in plot description can mislead the Chi-Square test. To solve this problem, we clean the corpus by removing the plot description from reviews, before giving it as an input to the Chi-Square test. Cleaning of the corpus is done automatically by finding patterns for plot description in movie reviews. In this paper, we perform the Chi-Square test with adjectives extracted from both cleaned and non cleaned corpus. The orientation of polarity of the output sentiment words are predicted simultaneously.

#### 3.1 Sentiment Word Extraction and Polarity Assignment

The key idea is that if a word does not belong to a particular class, then it should be uniformly distributed among all classes. So, before starting the test, we consider a null hypothesis. A null hypothesis states that if a given word is neutral, its chance to occur in positive and negative documents is equal. The value of a null hypothesis is equal to the arithmetic mean of the word count in positive and negative documents. This can also be considered as the expected count of words in both the classes of documents. We apply the Chi-Square test on the expected count and the actual observed count of the word. Deciding the polarity of words, that are used very rarely in corpus, is not worth considering. Since, if a word is polar, then it will occur frequently in polar documents. So we give only those words as input to the Chi-Square test, whose mean value is greater than 6. If the Chi-Square test results in a value, which is

greater than the threshold value, then there is a significant difference between the expected and the observed count of the word. At this moment we reject the null hypothesis and consider the possibility that there is some other factor causing the observed count to differ from the expected count of words. This factor is nothing but the polarity of adjectives, which makes it appear in a particular type of documents, frequently. If the word has positive sentiment, then it will occur more frequently in positive documents. Consider the following example.

*mesmerizing, unpredictable, thrilling, non-stop*

Negatively polar words occur more frequently in negative documents. Consider the following example.

*juvenile, predictable, underwritten, murky*

The extraction approach is best described in Algorithm 1.

The Bidirectional Stanford POS tagger<sup>7</sup> is used to tag words from the corpus with parts of speech. Experiments are performed with different thresholds for the Chi-Square value of the adjective.

#### 3.2 Cleaning of Corpus

In the movie domain, reviewers feel free to describe the plot of the movie as part of the review for a better understanding of it. So, in the movie domain, the *cleaning of the corpus* is mandatory because the polar words which are present in the plot part may mislead the classifier. However, cleaning of the corpus is not required in other domains, for example, *Camera* and *Cell Phones*. We find patterns that represent plot description in the corpus.

- Some reviewers have explicitly divided reviews into two parts - one for review and another for the movie plot - under different titles. It is shown in table 1.
- Some reviewers have specified that the *review contains spoilers*.
- We are performing experiments with the *English* movie review corpus, so movie names

<sup>5</sup>[www.rottentomatoes.com](http://www.rottentomatoes.com), [www.imdb.com](http://www.imdb.com)

<sup>6</sup>Available at [www.cs.cornell.edu/people/pabo/movie-review-data/](http://www.cs.cornell.edu/people/pabo/movie-review-data/) (review corpus version 2.0)

<sup>7</sup><http://nlp.stanford.edu/software/tagger.shtml>

**Input:** Domain Specific Corpus Tagged with POS

**Output:** Sentiment Lexicon with Polarity

**foreach** *WORD* in the corpus **do**

**if** POS of *WORD* is JJ or JJS **then**

    T:= get total count(*WORD*)

    P:= get count in positive documents(*WORD*)

    N:= get count in negative documents(*WORD*)

    Expected\_Count := T/2;

**if** Expected\_Count > 6 **then**

$Chi^2(WORD) := \frac{((P - Expected\_Count)^2 + (N - Expected\_Count)^2)}{Expected\_Count}$

**if**  $Chi^2 > Threshold$  **then**

**if** (P - N) > 0 **then**

          Polarity := +1;

          Add\_To\_Sentiment\_Lexicon(*WORD*,Polarity);

**else**

          Polarity := -1;

          Add\_To\_Sentiment\_Lexicon(*WORD*,Polarity);

**else**

        Continue for next *WORD*;

**else**

      Continue for next *WORD*;

**else**

    Continue for next *WORD*;

**end**

**Algorithm 1:** Extraction of sentiment lexicon with the polarity

may overlap with adjectives, for example, *unhappy birthday*, *13th warrior*. In a few places in reviews, movie names are given inside *double quotes*.

We find such files that match the pattern described above automatically and delete the found pattern.

## 4 Experimental Setup and Discussion

We use customer review collection as input data. The collection contains 1000 positive reviews and 1000 negative reviews. Experiments are done with cleaned and non cleaned corpora. We perform experiments with three threshold values 1.07, 2.45, 3.84. The threshold value specifies the minimum

Plot Part	Review Part
Plot	Critique
Synopsis	Comment
Synopsis	Reviews
Ingredient	Opinion

Table 1: Parts of a Review

probability<sup>8</sup> to accept a null hypothesis. For example, a threshold value of 1.07 indicates that there must be more than a 30% probability, to accept a null hypothesis. If the Chi-Square value of a word is greater than 1.07, we can conclude from the Pearson Chi-Square probability table that there is less than 30% probability, to accept a null hypothesis. Hence, reject the null hypothesis and consider word as candidate for sentiment lexicon.

1.07 also classifies boundary words, whose sentiment is not very clear from the corpus. Boundary words are those words that have almost equal occurrence in positive and negative documents, since they occur less frequently in the whole corpus. So such words fail to qualify the Chi-Square test with higher threshold values, but are actually polar. With threshold values, 2.45, 3.84 we get an increment in precision at the cost of leaving some boundary words unclassified.

In one of the experiments, we were able to retain words with poor Chi-Square value and higher threshold, that is-3.84, with the help of universal sentiment lexicon. Universal sentiment lexicon contains words which are strongly polar independent of the domain(Wilson et al., 2005). If a word has been rejected by the Chi-Square test with a threshold of 3.84, and it belongs to universal sentiment lexicon, then the correct polarity of the word can be derived from universal sentiment lexicon. Consider the following examples.

*Distracting* gets a Chi-Square value 2.0 but certainly negative in all domains.

*Monotonous* gets a Chi-Square value 2.25 but certainly negative in all domains.

## 5 Results and Discussion

Since there is no gold standard sentiment lexicon for the movie domain, the quality of output obtained through the Chi-Square test is confirmed by the inter annotators agreement. We ex-

<sup>8</sup><http://faculty.southwest.tn.edu/jwilliams/probab2.gif>

tracted 11,828 adjectives from corpus as candidates for lexicon. Among them 932 adjectives fulfill the Chi-Square test on non-cleaned corpus with threshold 1.07. **476 adjectives** are marked as true positives by **inter annotators agreements**. Table 2 shows the precision obtained with non cleaned corpus. With a threshold of 1.07, we get a precision of 51%. This result is affected by the words that occur in the plot description. Words which are part of the plot description mislead the classifier, causing low precision. Table 3 shows an improvement in precision with the cleaning of the corpus. The Chi-Square test with a threshold of 3.84, and universal sentiment lexicon gives a very high precision, that is 69.1%.

With a small threshold of 1.07, we are able to fetch almost all the words from the corpus that can be candidates for sentiment lexicon in the movie domain. With this intuition, we use true positives (476) and false positives (456) extracted by the Chi-Square test with threshold of 1.07 on the non cleaned corpus as a gold standard data to calculate recall and accuracy for experiments whose results are shown in table 3.

Data Set	Threshold	Precision
Non-Cleaned Corpus	1.07	51.07%
Non-Cleaned Corpus + Universal Sentiment Lexicon	3.84	69.1%

Table 2: Precision of the Chi-Square test with a non-cleaned Corpus

Table 3 shows results of precision, recall with increasing threshold values for the Chi-Square test.

Threshold	Precision	Recall
1.07	54%	100%
2.45	59%	82%
3.84	61%	65%

Table 3: Precision of the Chi-Square test with a cleaned Corpus

Table 3 shows that, as the value of the threshold increases, the precision increases. However, recall decreases. Figure 1 shows the accuracy obtained with different Chi-Square threshold values. The words which have a good Chi-Square score

are strong candidates for sentiment lexicon. But the words which are actually polar in the movie domain, but have been used very occasionally by reviewer, get rejected with an increase in the value of the threshold.

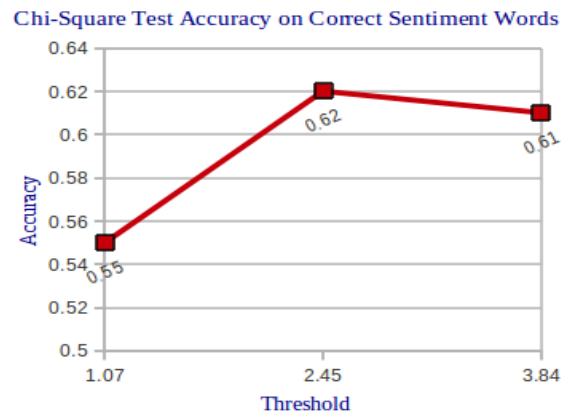


Figure 1: Chi-Square Test Accuracy with Different Thresholds

From figure 1, we can observe that accuracy is highest with a threshold of 2.45. When we move towards a higher threshold values, accuracy starts decreasing because of the higher fall in recall.

## 6 Conclusion

In this paper, we proposed a scheme to detect domain-dedicated sentiment words from the corpus. Our algorithm identifies polar words through an innovative application of Chi-Square test on the difference in the counts of the word in positive and negative documents. We extract a list of words that are polar in the movie domain, but cannot be in a universal sentiment lexicon. Our work is important because without incorporation of such domain specific polar words, the recall of a sentiment analysis system deteriorates. Experimental results show that our proposed method is promising and can be implemented for any domain. Our future work will focus on improving the precision by incorporating the effects of conjunction and negation.

## References

- Alex Cheng and Oles Zhulyn. 2012. "A System For Multilingual Sentiment Learning On Large Data Sets". Proceedings of the 24th International Conference on Computational Linguistics.
- Andrea Esuli and Fabrizio Sebastiani. 2005. "Determining the semantic orientation of terms through

- gloss classification*". Proceedings of the 14th ACM international conference on Information and knowledge management, 617–624.
- Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 1997. "*Thumbs up?: sentiment classification using machine learning techniques*". Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 79–86.
- Eric Breck, Yejin Choi and Claire Cardie. 2007. "*Identifying expressions of opinion in context*". Proceedings of the 20th international joint conference on Artificial intelligence, 2683–2688.
- Guang Qiu, Bing Liu, Jiajun Bu and Chun Chen. 2009. "*Expanding domain sentiment lexicon through double propagation*". Proceeding of 21st International joint conference on Artificial intelligence, 1199–1204.
- Hiroshi Kanayama and Tetsuya Nasukawa. 2006. "*Fully automatic lexicon expansion for domain-oriented sentiment analysis*". Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 355–363.
- Hong Yu and Vasileios Hatzivassiloglou. 2003. "*Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences*". Proceedings of the 2003 conference on Empirical methods in natural language processing, 129–136.
- Janyce Wiebe. 2000. "*Learning Subjective Adjectives from Corpora*". Proceedings of the Seventeenth National Conference on Artificial Intelligence, 735–740.
- Minqing Hu and Bing Liu. 2004. "*Mining and summarizing customer reviews*". Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 168–177.
- Peter D Turney. 2002. "*Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews*". Proceedings of the 40th annual meeting on association for computational linguistics, 417–424.
- Theresa Wilson, Janyce Wiebe and Paul Hoffmann. 2005. "*Recognizing contextual polarity in phrase-level sentiment analysis*". Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, 347–354.
- Vasileios Hatzivassiloglou and Katherine R McKeown. 1997. "*Recognizing contextual polarity in phrase-level sentiment analysis*". Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, 174–181.

# Can I hear you? Sentiment Analysis on Medical Forums

**Tanveer Ali**  
University of Ottawa  
tali028@uottawa.ca

**David Schramm**  
University of Ottawa and CHEO  
dschramm@ottawahospital.on.ca

**Marina Sokolova**  
University of Ottawa and CHEO  
sokolova@uottawa.ca

**Diana Inkpen**  
University of Ottawa  
Diana.Inkpen@uottawa.ca

## Abstract

Text mining studies have started to investigate relations between positive and negative opinions and patients' physical health. Several studies linked the personal lexicon with health and the health-related behavior of the individual. However, few text mining studies were performed to analyze opinions expressed in a large volume of user-written Web content. Our current study focused on performing sentiment analysis on several medical forums dedicated to Hearing Loss (HL). We categorized messages posted on the forums as *positive*, *negative* and *neutral*. Our study had two stages: first, we applied manual annotation of the posts with two annotators and have 82.01% overall agreement with kappa 0.65 and then we applied Machine Learning techniques to classify the posts.

## 1 Introduction

Natural language statements can be divided into two categories: factual and emotional. Factual statements can be expressed with a few topic keywords, while emotional statements express sentiments of the statement's author and require a more complex analysis than the factual ones.

Sentiment Analysis is often regarded as classifying and identifying the subjective information in the natural language text. In its application, Sentiment Analysis aims to detect the sentiments (e.g., opinions and emotions) of the speaker of the statement. Sentiments are characterized by polarity, intensity, strength and immediacy.

In the current study, we focus on the polarity of sentiments that are expressed in messages posted on medical forums. Polarity can be binary (e.g., positive vs. negative) or multi-categorical (e.g., positive, negative and unknown). Below we list examples found in online discussions about hearing aids.

## Positive<sup>1</sup>

*This has the beneficial effect of making the quieter sounds audible but not blowing your head off with the louder sounds.*

## Neutral/Unknown

*Now, you'll hear some people saying that compression is bad, and linearity is good, especially for music.*

## Negative

*Someone with 50 DB hearing aid gain with a total loss of 70 DB may not know that the place is producing 107 DB since it may not appear too loud to him since he only perceives 47 DB.*

In this work, we classified the subjective sentences into positive, negative and neutral. We have identified different syntactic features, i.e., patterns / rules (Yi and Nasukawa, 2003) which can indicate subjectivity and polarity of the sentences. The dataset of 3515 sentences from 26 threads were manually annotated by two annotators having overall agreement of 82.01% and kappa 0.65 which indicates substantial agreed data.

Our experiments with different combinations of features using different classifiers have shown significant improvement in performance over the baseline. For example, with the Naïve Bayes classifier, the F1-score was 10.5% better.

The rest of the paper is organized as follows: we discuss the sentiment analysis of health-related online messages, then we introduce our data; next we discuss the Subjectivity Lexicon and the features we use to represent the data, the analysis of the manual annotation and the machine learning classification results, before we conclude the presentation.

---

<sup>1</sup>All textual examples keep the original spelling and grammar.



## 2 Related Work

Very little work has been done in sentiment analysis on health-related forums. In (Goeuriot et al., 2012), the authors have built a medical domain lexicon in order to perform classification on a dataset that they collected from a website called Drug Expert. The dataset contains user reviews on drugs with ratings from 0 to 10 (Negative to positive) and they achieved F-score of 0.62 for the positive class, 0.48 for the negative class and 0.09 for the neutral class. The authors have performed the polarity detection on this dataset which already contains subjective information (opinions) about users' experience with particular drugs. However, in our case, we have extracted messages from health forums which contain mixed subjective and non-subjective information.

Users express their sentiments differently on forums compared to the way they express opinions when providing reviews or sharing messages on social networks. Bobicev et al. (2012) have analyzed sentiments in Twitter messages using some statistical features based on the occurrence and correlation among words with the class labels of the training set. However, we have identified the correlation of phrases within sentences for predicting subjectivity and polarity.

## 3 Building the Dataset

Surgeries related to HL are the most common surgeries in North America; thus, they affect many patients and their families.

However, there are only a few health forums dedicated to Hearing Loss (HL). Hence, we did not have an access to a high volume of data. Also, we need forum discussions, i.e., threads, which consist of more opinionated messages rather than questions and answers about the medical problems.

For the sentiment analysis, we have chosen a critical domain of HL problems: opinions about Hearing Aids. To the best of our knowledge, no relevant previous work was done in this area. For our dataset, we have collected individual posts from 26 different threads on three health forums<sup>2</sup>.

### 3.1 Data Description

The initial collection of data contains about 893 individual posts from 34 threads. They were

extracted using the XPath query by using the Google Chrome extension "XPathHelper".

This data was filtered and reduced to 607 posts in 26 threads (Table 1), by removing the threads where people discussed the factual information about a specific problem or disease and which do not contain any sentiments or opinions. Statistics, like average posts per person, were measured for filtering the data. For example, threads with more than 100 posts were removed, as threads with a large number of posts deviated from the main topic of discussion.

	Threads	Posts	Avg. posts per person
www.hearingaidforums.com	7	185	2.9
www.medhelp.org	9	105	2.77
www.alldeaf.com	10	317	1.93
Total	26	607	2.53

**Table 1. Filtered dataset collection statistics**

We split the data from individual threads into sentences using our version of a regular expression-based sentence splitter. We partly removed noise from the text by removing sentences containing very few words (i.e., less than 4 in our case). The remaining sentences from the 26 threads were manually annotated by two independent annotators into three classes (Positive, Negative and Neutral/Unknown).

## 4 Subjectivity Lexicon

For our experiments, we used the Subjectivity Lexicon (SL) built by Wilson, Wiebe, and Hoffman (2005). The lexicon contains 8221 subjective expressions manually annotated as strongly or weakly subjective, and as positive, negative, neutral, or both. We have chosen this lexicon over other large automatically-generated dictionaries like SentiWordNet (Baccianella, Esuli, and Sebastiani, 2010), as it has been manually annotated and provides rich information with the subjectivity strength and prior polarity for each word considering the context of the word in the form of part of speech information.

The quality of this Subjectivity Lexicon is higher than the quality of other large automatically generated dictionaries; for example, SentiWordNet includes more than 65,000 entries. Some papers (Taboada et al., 2011) have shown that larger dictionaries contain information which is not detailed and include more words which may lead to more noise.

<sup>2</sup> <http://www.medhelp.org>, <http://www.alldeaf.com>, <http://www.hearingaidforums.com>

Below is a sample entry from the lexicon:

*type=weaksubjen=1 word1=ability pos1=noun stemmed1=n priorpolarity=positive*

This entry contains the term *ability*, which is a noun. Its length is 1 (single term); it is not stemmed; it is weakly subjective and positive.

	Posi- tive	Nega- tive	Neu- tral	Bot h	Total	Per- cent
Adjec- jec- tive	1171	1838	235	5	3249	39.52
Noun	677	1346	144	3	2170	26.40
Verb	380	869	68	8	1325	16.12
any- pos	362	676	104	5	1147	13.95
Ad- verb	128	183	19	0	330	4.01
Total	2718	4912	570	21	8221	100
Per- cent	33.06	59.75	6.93	0.26	100	

**Table 2. Distribution of prior polarities within Subjectivity Lexicon**

The Subjectivity Lexicon contains only single term expressions. Table 2 shows that about 60% of the words are negative and 33% are positive. Also, this resource contains 40% adjectives, 26.4% nouns, 16.12% verb, 13.95% anypos (could be in any part of speech) and only 4% adverbs. Table 3 shows that about 67.74% of the words are strong subjective and the rest of 32.2% are weak subjective in nature.

	Strong Subj	Weak Subj	Total	Percent
Adjective	2006 (61.74%)	1243 (38.25%)	3249	39.52
Noun	1440 (25.85%)	730 (33.6%)	2170	26.40
Verb	861 (15.46%)	464 (35.01%)	1325	16.12
Anypos	1043 (18.72%)	104 (9.06%)	1147	13.95
Adverb	219 (3.93%)	111 (33.6%)	330	4.01
Total	5569	2652	8221	100
Percent	67.74	32.26	100	

**Table 3. Distribution of subjectivity clue within the Subjectivity Lexicon**

The lexicon contains only 21 words having polarity “both”. Out of these 21, only 10 words

were found unique with their part of speech. As these both polarity words are neutral in our case, we decided to merge them with the neutral words. Table 4 shows the relation between strong and weak subjectivity with the polarity lexicon.

	Strong Subj	Weak Subj	Total	Percent
Positive	1717 (30.8%)	1001 (37.74%)	2718	33.06
Negative	3621 (65%)	1291 (48.6%)	4912	59.75
Neutral	231 (4.14%)	360 (13.57%)	591	7.18
Total	5569	2652	8221	100
Percent	67.74	32.26	100	

**Table 4. Distribution among subjectivity and polarity in the lexicon**

## 5 Methodology

In this work, we have used several different features for the sentiment analysis of the sentences. Section 4.2 lists all these features. These features are computed and presented for each sentence in a data file format used by the WEKA tool (Hall et al., 2009). Classification is performed based on the computed features and accuracy is measured using different combinations of features in order to improve the classification performance.

### 5.1 Parts of Speech in Lexicon Matching

Words can have different polarity when they represent different parts of speech; e.g., novel is positive when it is in adjective form; however it is a neutral as a noun. To minimize this problem, we have matched the words in the lexicon with their part-of-speech information. That helped us to use the correct polarity and subjectivity indication considering the correct part of speech.

#### Nouns

In our lexicon, nouns have the second most coverage, with 26.4%.

#### Verbs

Verbs are the next common in the lexicon and give good indication of subjectivity. However, as verbs are used in many different forms and have many meanings, just relying on the verb polarity will misguide the prediction in cases where the verbs are used in some other senses, e.g., *he uses a car* is neutral, when *he was used* has a negative sense.

## Lemmatization

For all nouns and verbs, we have used the lemmatization from the GATE<sup>3</sup> morphological plugin, which provides the root word. In case of nouns, the root word is the singular form of the plural noun, e.g., *bottles* become *bottle*, etc. In the case of verbs, the plugin provides the base form for infinitive, e.g., *helping* becomes *help*, and *watches* become *watch*. After performing lemmatization, we found 158 more words that were detected with the same part of speech considered as the original. There were still 175 words which were found with the root word in the lexicon, but with different part of speech, e.g., *senses* was used as noun in the data; after lemmatization it becomes *sense*, which exists as verb in the lexicon. Therefore it cannot be matched, as the context and meaning of the word is different.

## Adjectives

Early research in sentiment analysis focused mainly on adjectives and phrases containing adjectives, e.g., *what a blessed relief*. Adjectives are good indicators for the positivity or negativity of the sentences, but they are not sufficient for identifying the subjectivity in the sentences, as we will see in the experiments.

## Adverbs

Adverbs are words that modify the verbs, adjectives and other phrases or clauses, e.g., *I am usually a contributing adult, and am happily sane and I say whoa how did that happen?* Adverbs have the lowest concentration in the lexicon, only 4%, and as many adverbs are identified by their characteristic "ly" suffix, we have removed the suffix-ly and then matched the new word in the lexicon by considering it as adjective. In English, most of the adverbs with suffix -ly such as *badly, softly, carefully, extremely* are forms of adjectives; therefore considering these provides better results in predicting the polarity of words in their correct senses.

## Features

All the features considered for the experiment are based on sentence level. Table 5 shows the final features selected for the experiments. The most common features were pronouns, followed by weak subjective clues, adjectives, and adverbs. There were more words that matched with the lexicon's positive words than those that

matched with the lexicon's negative words. This led to classifier's performance become slightly better for positive in the experiments.

STRONGSUBJ	# of words found as strong subjective in current sentence
WEAKSUBJ	# of words found as weak subjective in current sentence
ADJECTIVE	# of adjectives
ADVERBS	# of adverbs
PRONOUN	# of pronouns
POSITIVE	# of words found having prior polarity as positive
NEGATIVE	# of words found having prior polarity as negative
NEUTRAL	# of words found having prior polarity as neutral
PRP_PHRASE	# of phrases containing pronouns found in current sentence

**Table 5. Final features considered for the experiments**

## 6 Sentiment Categories

The dataset of 3515 sentences from 26 threads were manually annotated by two annotators. The annotators were asked to tag each sentence into positive, negative and neutral (where both positive and negative sentiments are discussed). All the sentences which do not contain any opinions are left blank and they are removed, as we focus on sentences containing sentiments. According to Table 6, annotator1 and annotator2 did not label a large number of sentences, i.e., 2939 and 2728, respectively; therefore these sentences are removed. Due to the large number of unlabeled sentences, the data is reduced, as we consider only those sentences labeled as positive, negative and neutral. Since the positive and negative dataset is already balanced, no data balancing is performed.

Annotator 2	Annotator 1				Total
	Pos	Neg	Neut	No Label	
Pos	226				329
Neg		214			296
Neutral			117		162
No Label				2720	2728
Total	230	218	128	2939	3515

**Table 6. Annotations statistics of Sentences between the two annotators**

<sup>3</sup><http://gate.ac.uk/sale/tao/splitch21.html#x26-52600021.11>

The overall agreement for the two datasets is computed through the commonly used kappa statistic to evaluate the agreement ratio between the two annotators, in the same form used in (Sokolova & Bobicev, 2011):

$$\text{kappa} = \frac{\frac{a+d}{N} - \frac{f_1g_1+f_2g_2}{N^2}}{1 - \frac{f_1g_1+f_2g_2}{N^2}}$$

The overall percentage agreement between the annotators for the positive/negative dataset was 82.01% and kappa was 0.65. This indicates a substantial agreement between the taggers.

Positive / Negative dataset									
	Naïve Bayes			SVM			Logistics-R		
	P	R	F-1	P	Re	F-1	P	R	F-1
<b>positive, negative</b>	0.656	0.65	<b>0.644</b>	0.661	0.641	<b>0.625</b>	0.649	0.645	<b>0.641</b>
all features	0.595	0.584	0.565	0.641	0.618	0.596	0.657	0.657	<b>0.656</b>
Baseline	0.540	0.541	0.539	0.586	0.586	0.586	0.585	0.584	<b>0.584</b>

**Table 7. Comparison of performance between different features among three classifiers for both datasets**

Positive / Negative dataset with lemmatization									
	Naïve Bayes			SVM			Logistic-R		
	P	R	F-1	P	Re	F-1	P	R	F-1
positive, negative	0.644	0.625	<b>0.607</b>	0.636	0.607	<b>0.578</b>	0.688	0.686	<b>0.685</b>
all features	0.589	0.580	0.560	0.627	0.600	0.570	0.671	0.670	0.670
Baseline	0.540	0.541	0.539	0.586	0.586	0.586	0.585	0.584	<b>0.584</b>

**Table 8. Comparison of performance with lemmatization between different features among three classifiers for both datasets**

## 7 Experiments

The output files generated by the system for the dataset are classified using the WEKA tool (Hall et al., 2009). For our evaluation, we used 10-fold cross validation which is a standard classifier selection for classification purpose. Experiments were performed using three different classifiers: Naïve Bayes, because it is known to work well with text, support vector machine (SVM) because of its high performance in many tasks, and logistic regression (logistic-R), in order to try one more classifier based on a different approach.

Performance was evaluated using the F1-measure between the three classifiers on the given datasets. We found that the performance of logistics regression was the best on the features selected for our evaluation.

For the baseline, the feature vector of bag of words is considered for both the datasets. We have not considered the unique words for the bag

of words because eliminating the words that appeared only once halves the size of the vectors, thus it makes it easier for the classifier to handle bag-of-words; also the unique words do not contribute much in classification since they appear only once, in one class. Table 7 shows significant improvement with positive, and negative features over the baseline and the difference was much higher with Naïve Bayes and SVM than with logistic-R, i.e., 10.5%, 3.9% and 5.7%, respectively.

In Table 8, for the positive/negative dataset, the classifiers Naïve Bayes and SVM underperformed with the lemmatization and their best performance decreased by 3.7% and 4.7%, respectively. However, the performance of logistic-R increased significantly, by 4.4%, and its F1-measure reached 68.5%, which indicates the benefit of lemmatization in matching within the lexicon.

## 8 Analysis

The results from the experiments have provided several insights about the sentiment analysis in health-related forums. Note that the bag-of-word representation (BOW) is a high baseline that is hard to beat in many texts classification problems. The Subjectivity Lexicon clues for polarity such as positive, negative and neutral have shown significant improvement for the identification of positive and negative sentences. As a result, the performance has increased by 4.2% on average among the three classifiers.

We have noticed that for the semantic orientation of the sentences, the combination of lexicon clue features with other basic counting features such as the number of adjectives, the number of adverbs, etc., decreased the performance of classification, as all the three classifiers have performed best with only positive and negative features.

Our results are comparative to other related studies for sentiment classification of medical forums. Sokolova & Bobicev (2011) achieved the best F-score of 0.708 using SVM; similarly Goeuriot et al. (2012) for drug reviews achieved F-score of 0.62 for the positive class, 0.48 for the negative class and 0.09 for the neutral class.

In general, for consumer reviews, opinion-bearing text segments are classified into positive and negative categories with Precision 56%–72% (Hu & Liu 2004). For online debates, the complete texts (i.e., posts) were classified as positive or negative stance with F-score 39%–67% (Somasundaran & Wiebe, 2009); when those posts were enriched with preferences learned from the Web, the F-score increased to 53%–75%.

It is also noted that the classification for semantic orientation depends heavily on the quality of the lexicon used, rather than the size of the lexicon, as the results show that the classification of the sentences into positive and negative reached 70% by using only the polarity clues for individual words within the lexicon.

## 9 Conclusion and Future Work

In this work, we performed the sentiment analysis of sentiments expressed in online messages related to Hearing Loss.

We used several lexicon-based features together with rule-based features like pronoun phrases for our classification of the dataset for detecting semantic orientation within the subjective data us-

ing different features based on the subjective lexicon.

The dataset of 3515 sentences from 26 threads were manually annotated by two annotators and achieved 82.01% overall agreement with kappa 0.65. Evaluations have been made for the classification of the substantial agreed data using three different supervised learning-based classifiers and it is shown that our proposed features outperformed the baseline of bag-of-word features by 10.5% with Naïve Bayes, 3.9% with SVM and 5.7% with logistic-R.

In future work, we could consider several directions. The lexicon could be improved, as the domain lexicon created in (Goeuriot et al., 2012) has shown better results over other dictionaries for polarity detection of the sentences.

Also, techniques and features presented in (Taboada et al., 2011), (Kennedy and Inkpen, 2006) such as intensification (e.g., *very good*) increase the polarity of *good* and negation (e.g., *not good*) which reverses the polarity of *good*, can be used for the semantic orientation or polarity detection of the sentences.

Another direction for future work could be to investigate changes of sentiments in threads. We want to analyze what linguistic events may prompt polarity to reverse (e.g., from *positive* to *negative*) and under what conditions the same polarity is sustained. To the best of our knowledge, this task was not addressed before.

## Acknowledgments

This work in part has been funded by a Natural Sciences and Engineering Research Council of Canada Discovery Research Grant and by a Children's Hospital of Eastern Ontario Department of Surgery Research Grant.

## References

- Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). *Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining*. In Proceedings of the 7th conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, May.
- Bobicev, V., Sokolova, M., Jafer, Y., & Schramm, D. (2012). *Learning sentiments from tweets with personal health information*. In Advances in Artificial Intelligence (pp. 37-48). Springer Berlin Heidelberg.

- Eysenbach, G. (2009). *Infodemiology and infoveillance: framework for an emerging set of public health informatics methods to analyze search, communication and publication behavior on the Internet*. *Journal of medical Internet research*, 11(1).
- Gillick, D. (2009, May). *Sentence boundary detection and the problem with the US*. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers (pp. 241-244). Association for Computational Linguistics.
- Goeuriot, L., Na, J. C., Min Kyaing, W. Y., Khoo, C., Chang, Y. K., Theng, Y. L., & Kim, J. J. (2012, January). *Sentiment lexicons for health-related opinion mining*. In Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium (pp. 219-226). ACM.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). *The WEKA data mining software: an update*. *ACM SIGKDD Explorations Newsletter*, 11(1), 10-18.
- Hu, M., & Liu, B. (2004, August). *Mining and summarizing customer reviews*. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 168-177). ACM.
- Kennedy, A., & Inkpen, D. (2006). *Sentiment classification of movie reviews using contextual valence shifters*. *Computational Intelligence*, 22(2), 110-125.
- Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M. (2003). *Lying words: Predicting deception from linguistic styles*. *Personality and Social Psychology Bulletin*, 29(5), 665-675.
- Rhodewalt, F., & Zone, J. B. (1989). *Appraisal of life change, depression, and illness in hardy and non-hardy women*. *Journal of Personality and Social Psychology*, 56(1), 81.
- Sokolova, M., & Bobicev, V. (2011). *Sentiments and Opinions in Health-related Web messages*. In Recent Advances in Natural Language Processing (pp. 132-139).
- Somasundaran, S., & Wiebe, J. (2009, August). *Recognizing stances in online debates*. 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-Volume 1 (pp. 226-234). Association for Computational Linguistics.
- Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). *Lexicon-based methods for sentiment analysis*. *Computational linguistics*, 37(2), 267-307.
- Wilson, T., Wiebe, J., & Hoffmann, P. (2005, October). *Recognizing contextual polarity in phrase-level sentiment analysis*. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (pp. 347-354). Association for Computational Linguistics.
- Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003, November). *Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques*. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on* (pp. 427-434).

# Construction of Emotional Lexicon Using Potts Model

**Braja Gopal Patra<sup>\*</sup>, Hiroya Takamura<sup>+</sup>, Dipankar Das<sup>#</sup>,  
Manabu Okumura<sup>+</sup> and Sivaji Bandyopadhyay<sup>\*</sup>**

<sup>\*</sup>Dept. of Computer Science & Engineering, Jadavpur University, Kolkata, India

<sup>#</sup>Dept. of Computer Science & Engineering, NIT Meghalaya, India

<sup>+</sup>Precision and Intelligence Laboratory, Tokyo Institute of Technology, Japan

brajagopal.cse@gmail.com, takamura@pi.titech.ac.jp,  
dipankar.dipnil2005@gmail.com, oku@pi.titech.ac.jp,  
sivaji\_cse\_ju@yahoo.com

## Abstract

Emotion is an instinctive state of mind aroused by some specific objects or situation. Exchange of textual information is an important medium for communication and contains a rich set of emotional expressions. The computational approaches to emotion analysis in textual data require annotated lexicons with polarity tags. In this paper we propose a novel method for constructing emotion lexicon annotated with Ekman's six basic emotion classes (anger, disgust, fear, happy, sad and surprise). We adopt the Potts model for the probability modeling of the lexical network. The lexical network has been constructed by connecting each pair of words in which one of the two words appears in the gloss of the other. Starting with a small number of emotional seed words, the emotional categories of other words have been determined. With manual checking of top 200 words from each class an average precision of 85.41% has been achieved.

## 1 Introduction

Sentiment analysis and classification from electronic text is a hard semantic disambiguation problem (Das and Bandyopadhyay, 2010). Many recent researches have been conducted in the fields of sentiment extraction (Kim et al., 2012; Taboada et al., 2011), opinion mining, summarization (Aman and Szpakowicz, 2007; Das et al., 2012; Yang et al., 2007; Quan and Ren, 2010) and each of which has a variety of potentially valuable applications. For example, we can efficiently collect people's opinion on a new rule

enforced by the Government from Blog sites and at the same time be able to grasp their emotion without having to read their comments. An imperative resource for such kind of emotional analysis is an emotion lexicon annotated with several emotional classes like *happy*, *sad*, *fear*, *anger*, *surprise* and *disgust*. In the previous example, frequent appearance of words from the *happy* class in a blog document would imply that the writer of the comment is quite happy with the new rule proposed by the Government.

Several works have been conducted on building emotional corpora in different languages such as in English (Aman and Szpakowicz, 2007), Chinese (Yang et al., 2007; Quan and Ren, 2010), and Bengali (Das and Bandyopadhyay, 2010) etc. All these works have focused on developing sentiment lexicon with three sentiment classes. For instance, Takamura et al. (2005) have developed a lexicon of emotion words tagged with the classes *desirable* and *undesirable* using Spin model. A number of other polarity sentiment lexicons are available in English such as SentiWordNet 3.0 (Esuli et al., 2010), Subjectivity Word List (Wilson et al., 2005), WordNet-Affect list (Strapparava et al., 2004), Taboada's adjective list (Taboada et al., 2006). On the other hand, several polarity sentiment lexicons have been developed in different languages like Hindi, Bengali and Telegu (Das and Bandyopadhyay, 2010), Japanese (Torii et al., 2012) etc.

Among all these publicly available sentiment lexicons, SentiWordNet is one of the well-known and widely used ones (number of citations is higher than other resources<sup>1</sup>), having been uti-

---

<sup>1</sup> <http://citeseerx.ist.psu.edu/index>

lized in several applications such as sentiment analysis, opinion mining and emotion analysis.

Undoubtedly, manual compilation is the best way to create such an emotion lexicon but is much expensive in terms of time and human effort. Thus, the objective of the present paper is to develop a method for automatically creating such a list of words from the glosses of a dictionary, as well as from a thesaurus and a corpus. For this purpose, we have used the *Potts model*, a probabilistic model for lexical network. In the lexical network, each node has one of the three orientation values and the neighboring nodes tend to have the same value. For each of the emotion classes, we estimate the states of the nodes indicating the semantic orientation of each class. However, the proposed method does not deal with words that do not appear in the lexical network.

We have classified the words into six emotion classes using Potts model. First, the manual evaluation has been done to get the accuracies. Then we have automatically calculated accuracies comparing with the WordNet Affect list. We have also classified the words into two classes (positive and negative) and the accuracy is evaluated using the SentiWordNet. The generated emotion lexicon in English also contains the parts of speech (adjective, adverb, noun and verb) information of the emotion words as well as their emotional classes.

The rest of the paper is organized in the following manner. Section 2 discusses briefly the resources available till date. Section 3 provides an overview on Potts model. Section 4 describes the implementation of Potts Model for the construction of our emotion lexicon. Section 5 presents the experiments with detail analysis. Finally, conclusions are drawn and future directions are presented in Section 6.

## 2 Related Work

Takamura et al. (2005) extracted semantic orientation of words according to the spin model, where the semantic orientation of words propagates in two possible directions like electrons. Electrons propagate their spin direction to neighboring electrons until the system reaches a stable configuration. They have constructed a lexical network by connecting pairs of words. In each pair either word appears in the gloss of the other. They have applied spin model iteratively till energy of the system is minimized.

Esuli and Sebastiani's (2006) approach to develop the SentiWordNet is an adaptation to synset classification based on the training of ternary classifiers for deciding positive and negative (PN) polarity. Each of the ternary classifiers is generated using the Semi-supervised rules.

Strapparava and Valitutti (2004) developed the WORDNET-AFFECT, a lexical resource that assigns to a number of WORDNET synsets one or more affective labels such as emotion, mood, trait, cognitive state, physical state, behavior, attitude and sensation etc. They have prepared a preliminary resource named as AFFECT, then projected part of the affective information from the AFFECT database onto the corresponding senses of WORDNET-AFFECT.

Das and Bandyopadhyay, (2010) created the SentiWordNet for Indian Languages like Hindi, Bengali and Telegu by multiple computational approaches like WordNet based, dictionary based, corpus based or generative approaches. They have used the Bilingual corpus and generated the SentiWordNet(s) for the Indian languages from the English sentiment lexicon merged from the English SentiWordNet and the Subjectivity Word List.

Das et al., (2012) presented a task of developing an emotion lexicon. A lexical network has been developed on the freely available ISEAR dataset using the co-occurrence threshold. They classified words into seven categories, i.e., *anger, disgust, fear, guilt, joy, sadness* and *shame*. SVM and Fuzzy C-mean classifier have been used for the classification. They also computed the precision of top 100 words and reported 95% precision for seven emotion classes.

## 3 Potts Model

We have employed the Potts model which is a generalization of Ising model (Nishimori, 2001). If a variable has more than two values and there is no ordering relation between the values, such network is called a Potts Model (Wu, 1982). Potts model has been a subject of increasing research interest in the recent years. In this section we present the mathematics of Potts model. Potts model has been used in several applications such as extraction of semantic orientations of phrases from dictionary (Takamura et al., 2007).

It has been observed that the types of similarity or prior polarity scores do not completely solve the problem of classifying emotional words. In fact, finer details are revealed by so-called contextual polarity classification, because



the same textual content can be presented with different emotional slants (Grefenstette et al., 2005). For example, the word ‘succumb’ can trigger a mix of multiple emotions: ‘fear’ as well as ‘sad’. Considering word-wise emotion identification as a multi-label text classification problem, we deploy a Potts model based classification technique.

### 3.1 Introduction to Potts Model

Suppose a network of nodes and weighted edges is given. The states of the nodes are collectively represented by  $n$ . The weight between nodes  $i$  and  $j$  is represented by  $w_{ij}$ .

The energy function is represented as  $H(n)$ , which indicates the state of the whole network:

$$H(n) = -\beta \sum_{ij} w_{ij} \delta(n_i, n_j) + \alpha \sum_{i \in L} -\delta(n_i, a_i)$$

where  $\beta$  is a constant called *the inverse-temperature*,  $L$  is the set of the indices for the observed variables,  $a_i$  is the state of each observed variable indexed by  $i$ , and  $\alpha$  is a positive constant representing a weight on labeled data. Function  $\delta$  returns 1 if two arguments are equal to each other, 0 otherwise. The state is penalized if  $n_i$  ( $i \in L$ ) is different from  $a_i$ . Using  $H(n)$ , the probability distribution of the network can be represented as  $P(n) = \exp\{-H(n)\}/Z$ , where  $Z$  is a normalization factor.

However, it is computationally difficult to exactly estimate the state of this network. We resort to a mean-field approximation method. In the method,  $P(n)$  is replaced by factorized function  $\rho(n) = \prod_i \rho_i(n_i)$ . Then we can obtain the function with the smallest value of the variational free energy:

$$\begin{aligned} F(n) &= \sum_n P(n)H(n) - \sum_n -P(n) \log P(n) \\ &= -\alpha \sum_i \sum_{n_i} \rho_i(n_i) \delta(n_i, a_i) \\ &\quad -\beta \sum_{ij} \sum_{n_i, n_j} \rho_i(n_i) \rho_j(n_j) w_{ij} \delta(n_i, n_j) \\ &\quad - \sum_i \sum_{n_i} -\rho_i(n_i) \log \rho_i(n_i) \end{aligned}$$

By minimizing  $F(n)$  under the condition that  $\forall_i, \sum_{n_i} \rho_i(n_i) = 1$ , we obtain the following fixed point equation for  $i \in L$ :

$$\rho_i(n) = \frac{\exp(\alpha \delta(n, a_i) + \beta \sum_j w_{ij} \rho_j(n_j))}{\sum_m \exp(\alpha \delta(m, a_i) + \beta \sum_j w_{ij} \rho_j(m))}$$

The fixed point equation for  $i \notin L$  can be obtained by removing  $\alpha \delta(n, a_i)$  from above.

This fixed point equation is solved by an iterative computation. After the computation, we obtain the function  $\prod_i \rho_i(n_i)$ . When the number of classes is two, the Potts model in this formulation is equivalent to the mean-field Ising model (Nishimori, 2001).

## 4 Potts Model for Construction of Emotional Lexicon

In this section we describe the methodologies adopted to develop the emotional lexicon wherein words are classified into six emotional classes.

### 4.1 Constructing Lexical Networks

We have constructed a lexical network which has been termed as gloss network (Takamura et al., 2005). This network is developed by linking two words if one appears in the gloss of other word. Each link belongs to one of the two groups: same orientation links (SL) and different orientation links (DL). If at least one word precedes a negation word (e.g., not) in the gloss of the other word, the said link is considered as a different-orientation link. Otherwise the link is a same-orientation link. Lexical Network contains 88015 words collected from the dictionary. Statistics of the lexical network is shown in Table 1. Next, we assign weights  $W = (w_{ij})$  to links as follows:

$$w_{ij} = \begin{cases} \frac{1}{\sqrt{d(i)d(j)}} & (e_{ij} \in SL) \\ -\frac{1}{\sqrt{d(i)d(j)}} & (e_{ij} \in DL) \\ 0 & otherwise \end{cases}$$

where  $e_{ij}$  denotes the link between word  $i$  and word  $j$ , and  $d(i)$  denotes the degree of word  $i$ , which is actually the number of words linked with word  $i$ . Two words without a connection are regarded as connected by a link of weight 0.

Class	No. of words
Adjective	20497
Adverb	3751
Noun	55285
Verb	8482

Table 1. Statistics of Lexical network

We have also constructed another network, *the gloss thesaurus network (GT)*, by linking syno-

nyms, antonyms and hypernyms, in addition to the above linked words. Only antonym links are in DL.

We enhanced the gloss-thesaurus network with co-occurrence information extracted from corpus. Hatzivassiloglou and McKeown (1997) focused on conjunctive expressions such as “simple and well-received” and “simplistic but well-received”, where the former pair of words tend to have the same semantic orientation, and the latter tend to have the opposite orientation. Following their method, we connect two adjectives if the adjectives appear in a conjunctive form in the corpus. If the adjectives are connected by “and”, the link belongs to SL. If they are connected by “but”, the link belongs to DL. We call this network the *gloss-thesaurus-corpus network (GTC)*. We have used *gloss-thesaurus-corpus network* in our experiments.

## 4.2 Classification of Words

Takamura et al., (2007) used the Potts model for extracting semantic orientation of phrases (pair of adjective and a noun): positive, negative or neutral. In contrast to that, we have used the Potts model for identifying the emotional class (es) of a word.

We have used one seed word from each class to start with the experiment. Each seed word is assigned a class manually. We therefore estimate the state of nodes in the lexical network for each class of emotions. The only drawback is that, it could not assign any emotional class to a word which is not present in the lexical network. These words may be referred to as *unseen words*.

The reason of choosing Potts model over Ising model is that Ising model is helpful for modeling a system involving two classes only (i.e. positive and negative), whereas Potts model can be modeled for more than two classes.

## 5 Evaluation

We performed our experiments with different values of  $\beta$ , ranging from 0.5 to 1 with an interval of 0.1 and achieved best result for  $\beta = 0.9$ . We also performed experiments with different set of seed words. Fixed seed words are used with different  $\beta$  values. We prepared three lists of seed words containing 6, 12 and 18 words respectively. They were prepared by picking 1, 2 or 3 words from each emotional class respectively.

We have classified the words into six emotional classes with different seed words and dif-

ferent values of  $\beta$ . Then the accuracies were computed manually as well as using the WordNet Affect lexicon. We also classified the words into two classes, i.e. positive and negative. The accuracies of two classes were calculated using the SentiWordNet.

Classes (Manually checked)	Precision (in %age)
Happy (200)	80.0
Sad (200)	80.5
Surprise (200)	82.0
Angry (200)	92.5
Fear (200)	92.0
Disgust (200)	85.5
Average	85.41

Table 2. Precision (under manual checking of each class)

## 5.1 Classification Results

Before discussing the accuracy, we would like to make some interesting observations. There are some words that cannot be classified by the classifier, i.e., for these words the probabilities of each class is the same. The number of these words varies with the change of  $\beta$  values and the number of seed words. We also observed similar change in the number of words put into each class.

We have manually checked top 200 words from each class having highest probability and reported the precision in Table 2. We have achieved maximum precision of 92.5% and 92.0% *angry* and *fear* classes respectively. It has been observed that the *happy* class has lowest precision and is about 80%. The precisions of *sad*, *surprise* and *disgust* classes are 80.5%, 82% and 85.5% respectively. We also performed several experiments by changing the values of  $\beta$  and the varying the number of seed words. The highest precision is achieved with  $\beta = 0.9$  and number of seed word kept at 18, i.e., three words from each group.

We observed that the accuracy of *happy* class is low. The reason may be that many words in this class do not have any relation to happy class and such words are basically neutral words or tough words, i.e. these words do not contain any emotions. For example, “*handel*” and “*olivier*” are identified as *happy* words, whereas they do not have any relation with the *happy* class. We also observed that *happy* emotion class contains

	Happy	Sad	Surprise	Angry	Fear	Disgust	Neutral
Happy	160	0	4	0	0	0	36
Sad	0	161	1	13	8	15	2
Surprise	5	6	164	2	8	12	3
Anger	0	7	1	185	1	5	1
Fear	0	9	1	1	184	3	0
Disgust	0	11	2	12	3	171	1
Neutral	0	0	0	0	0	0	0

Table 3. Confusion matrix for manual precision checking.

Classes	Precision (in %age)
Happy	50.8
Sad	52.3
Surprise	46.8
Angry	56.0
Fear	51.4
Disgust	35.5
Average	48.8

Table 4. Precision of each class collected by WordNet-Affect.

Classes	Precision (in %age)
Happy	58.9
Sad	65.4

Table 5. Accuracy of each class based on SentiWordNet.

some words from the *surprised* class. For example, the word “*fortuitous*” means happening by a lucky chance. Another example is “*stunning*”, which is classified as happy class, but it belongs to surprise class. In case of *sad* word class, we found some words from *fear*, *angry* and *disgust* classes. *Angry* class comprises some words from *sad*, *fear*, *disgust* and *neutral* classes. For example, the word “*stink*” is classified as *sad* class where as it belongs to *disgust* class. It does not contain any word from *happy* class. *Fear* and *disgust* classes contain word from all other classes except *happy* class. The details can be found from confusion matrix given in the Table 3.

There are some words which could belong to more than two classes depending on the context/situation. For example, “*shiver*” falls under the class *sad* and *fear*. We have removed these words while calculating the accuracy of the system.

We have also cross checked the accuracies of our system using the WordNet Affect. Here we have classified the words in six emotion classes and the precision is computed by comparing with WordNet-Affect. As WordNet-Affect contains less numbers of emotion words, so we just checked top 100 words from each emotion classes and the precision is given in Table 4. The average precision calculated is 48.8%. This is due to the fact that WordNet-Affect contains less number of words.

We have also classified the words into two classes, i.e. positive and negative. Then we have computed the accuracy of the output using the SentiWordNet 3.0 (Esuli et al., 2010). Approximately 25000 words occur with same probability and those words are removed at the time of testing. The accuracy is given in the Table 5. We have achieved 58.9 % accuracy in positive class or happy class, whereas 65.4% in negative class or sad class.

A shortcoming of this system is that it cannot handle those words which are not present in the lexical network. Error occurs when a non-emotional word is assigned a class.

## 6 Conclusion and Future Work

A method has been proposed for extracting emotional orientations of words with high accuracy using Potts model. The major contribution in the task is to prepare the emotional lexicon.

There are several directions for future works. One of them is to incorporate the syntactic information. Since importance of each word in a gloss depends on its syntactic role, syntactic information in glosses should be useful for classification.

A single word could belong to multiple classes. So, the identification of those words and representing them in fuzzy classes is one of the crucial research goals to be achieved in future.

Reducing the number of words having same probability in each emotion classes may be another research work. New words that are not listed in the Lexical Network can be updated in later works.

Finally, we are of the opinion that the proposed model is applicable to other tasks in computational linguistics.

## Acknowledgement

The work reported in this paper is supported by a grant from the India-Japan Cooperative Programme (DST-JST) 2009 Research project entitled “Sentiment Analysis where AI meets Psychology” funded by Department of Science and Technology (DST), Government of India.

## References

- Amitava Das and Sivaji Bandyopadhyay. 2010. SentiWordNet for Indian Languages. In *Proceedings of the 8th Workshop on Asian Language Resources (ALR)*, August, pages 56-63.
- Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*. Springer Berlin Heidelberg, pages 196-205.
- Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWord-Net: A publicly available lexical resource for opinion mining. In *Proceedings of the Language Resources and Evaluation (LREC)*, pages 417-422.
- 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, pages 1083-1086.
- Changhua Yang, Kevin Hsin-Yih Lin and Hsin-Hsi Chen. 2007. Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, pages 133-136.
- Dipankar Das, Soujanya Poria and Sivaji Bandyopadhyay. 2012. A classifier based approach to emotion lexicon construction. In *Proceedings of the Natural Language Processing and Information Systems*. Springer Berlin Heidelberg, pages 320-326.
- Fa-Yueh Wu. 1982. The Potts Model. *Reviews of Modern Physics*, 54(1):235-268.
- Gregory Grefenstette, Yan Qu, James G. Shanahan, David A. Evans. 2004. Coupling niche browsers and affect analysis for an opinion mining application. In *Proceedings of RIAO* (Vol. 4, pp. 186-194).
- Hidetoshi Nishimori. 2001. *Statistical Physics of Spin Glasses and Information Processing*. Oxford University Press.
- Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 133-140.
- Hiroya Takamura, Takashi Inui and Manabu Okumura. 2007. Extracting Semantic Orientations of Phrases from Dictionary. In *Proceedings of the Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT,2007)*, pages 292-299
- Maite Taboada, Anthony Caroline and Voll Kimberly. 2006. Creating semantic orientation dictionaries. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, Genoa, pp. 427-432.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267-307.
- Mitra Mohtarami, Hadi Amiri, Man Lan, Thanh P. Tran, and Chew L. Tan. 2012. Sense Sentiment Similarity: An Analysis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1706-1712.
- Paul Ekman 1993. Facial expression and emotion. *American Psychologist*. 48(4):384-392.
- Seungyeon Kim, Fuxin Li, Guy Lebanon and Irfan Essa. 2012. Beyond Sentiment: The Manifold of Human Emotions. *arXiv preprint arXiv:1202.1568*.
- Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May.
- Theresa Wilson, Janyce Wiebe and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the HLT/EMNLP 2005*, Vancouver, Canada.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th ACL and 8th Conference of the EACL*, pages 174-181.
- Yoshimitsu Torii, Dipankar Das, Sivaji Bandyopadhyay and Manabu Okumura. 2011. Developing Japanese WordNet affect for analyzing emotions. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 80-86.

# Suicidal Tendencies: The Automatic Classification of Suicidal and Non-Suicidal Lyricists Using NLP

**Matthew Mulholland**

Montclair State University

Montclair, NJ, USA

*mulhollandm2@mail.montclair.edu*

**Joanne Quinn**

Montclair State University

Montclair, NJ, USA

*quinnj11@mail.montclair.edu*

## Abstract

Can natural language processing be used to predict the likelihood that a musician will commit or has committed suicide? In order to explore this idea, we built a corpus of songs that includes a development set, a training set, and a test set, all consisting of different lyricists. Various vocabulary and syntactic features were then calculated in order to create a suicide/non-suicide song classifier. The features were input into the Weka machine learning suite and tested with an array of algorithms. We were able to achieve up to a 70.6% classification rate with the SimpleCart algorithm, a 12.8% increase over the majority-class baseline. Our findings suggest that syntactic and vocabulary features are useful indicators of the likelihood that a lyricist will commit or has committed suicide.

## 1 Introduction

Recently, research into the application of NLP to the detection of health illnesses has proved fruitful. For instance, in their study of the effects of dementia on writers, Le et al. (2011), guided by previous research, explored various hypotheses in the novels of three British writers. Their research found a decline in the type-token ratio of the novelists suffering from dementia. The use of passive constructions was also explored since it is generally believed that it represents a syntactic structure that is particularly complex and likely would be used less often by writers suffering from dementia. Indeed, they found that authors with dementia use less passives than their healthy peers. Their results indicate the potential for natural language processing in the language of mental illness. Similarly, much research into the application of NLP to depression and suicide prediction has been con-

ducted in recent years (Pestian et al. 2012; Sohn et al. 2012).

While one might not expect depression or a suicidal tendency to affect language in the same way as an illness such as dementia, it is reasonable to assume that there will be textual indications of these mental illnesses also. In Stirman and Pennebaker (2001), word use is treated as an indicator of the mental states of suicidal and non-suicidal poets. Stirman & Pennebaker developed the Linguistic Inquiry and Word Count program to analyze over 70 language dimensions, including: polarity, affect states, death, sexuality, tense, etc. Their research found a correlation between the likelihood of a poet committing suicide and his/her disengagement from society based on the suicidal poets' heavy use of first-person singular pronouns and decreased use of the first-person plural pronouns. Interestingly, they also noted that poets who had committed suicide generally used more sexual words and references to death than their non-suicidal counterparts. Additionally, latent semantic analysis has been used to detect depression in free texts (Neuman et al. 2010). By creating a semantic field from the words commonly associated with the concept of depression, Neuman et al. were able to accurately identify depressed people through their writing. Finally, other pertinent research involves the use of concrete nouns and the lack of abstract concepts in professionally-written poetry (Kao & Jurafsky, 2012).

Based on the prior research, we anticipated that the suicidal lyricists' use of first-person singular pronouns would differ from that of the non-suicidal lyricists, possibly being significantly higher. We also hypothesized that certain features like the usage of sensual and morbid words and the passive construction would be more prevalent in the works of suicidal lyricists. Additionally, we were interested in exploring the differences among other features, such as TTR, the degree to which

a text is polar and in which direction, the n-gram profiles of songs in relation to the n-gram profiles of suicide/non-suicide lyricists in general, and the semantic fields of other target emotions and affect states of the two groups.

## 2 Methods

First, a corpus of songs by male suicide and non-suicide lyricists was constructed, which consists of training, development, and test sets. The training set is comprised of 533 songs, of which 253 were written by four lyricists who have not committed suicide and 280 by five lyricists who did commit suicide. The test set consists of 63 songs by 5 non-suicidal lyricists and 46 songs by 4 suicidal lyricists. Finally, the development set consists of 168 songs, 94 from 5 non-suicidal lyricists and 74 from 6 suicidal lyricists. Finally, the test set contains 63 songs written by five non-suicide artists and 46 songs written by 4 suicide artists. Table 1 displays the composition of the sets.

In our search for lyricists who committed suicide, we looked for lyricists who met the following prerequisite: the suicide had to be relatively unambiguous. This requirement constrained the size of the corpus a great deal. In addition, we attempted to distribute the lyricists across the different sets in an even manner such that each set would be comprised of lyricists of a range of times and nationalities. For these reasons, it was a source of difficulty trying to create sets with unique lyricists. Although we did not require that each song be solely written by the lyricist in question due to the fact that it is often murky concerning to whom the lyrics should be attributed, we did make an attempt to exclude songs written entirely by bandmates or other musicians. Due to the lack of female lyricists who committed suicide, we were forced to consider exclusively male lyricists.

48% of the non-suicide corpus consists of songs written by one artist, Bob Dylan. Removal of 35% of those Dylan songs (and thus evening out the distribution of songs) did not significantly alter the classifier’s results. The other non-suicide lyricists contributed between 32-45 songs each. The training set of suicide songs is slightly over-represented by Elliott Smith (about 33% of the songs). The four remaining suicide artists each contributed between 11% and 20% of all songs in this set. We additionally used a development set of five non-suicide lyricists (94 songs) and six suicide lyricists

(74 songs), which was used to compute n-gram features.

Each song considered was searched for in on-line lyrics databases and was cleaned by hand, the lines being joined into punctuated sentences or phrases so that a POS-tagger and lemmatizer could be used. The lyrics were then tokenized using the OpenNLP tokenizer and lemmatized. Features were computed using Python and with some help from the UAM corpus tool (O’Donnell 2008)(which uses the Stanford Parser), especially for grammatical analysis.

Lyricist	Suicide	Set	Songs
Bob Dylan	n	train	123
Bob Marley	n	train	42
Mike Ness	n	train	43
Trent Reznor	n	train	45
Elliott Smith	y	train	99
Ian Curtis	y	train	40
Kurt Cobain	y	train	53
Pete Ham	y	train	34
Phil Ochs	y	train	54
<b>Total</b>		<b>train</b>	<b>533</b>
Ben Folds	n	test	11
Chris Bell	n	test	12
John Lennon	n	test	20
Neil Young	n	test	10
Paul Simon	n	test	10
Doug Hopkins	y	test	4
Peter Bellamy	y	test	18
Richard Manuel	y	test	10
Tom Evans	y	test	14
<b>Total</b>		<b>test</b>	<b>109</b>
Beck Hansen	n	dev	10
George Harrison	n	dev	24
Johnny Cash	n	dev	22
Thom Yorke	n	dev	13
Tom Petty	n	dev	25
Adrian Borland	y	dev	27
Darby Crash	y	dev	8
Jim Ellison	y	dev	12
Mel Street	y	dev	5
Michael Hutchence	y	dev	3
Stuart Adamson	y	dev	19
<b>Total</b>		<b>dev</b>	<b>168</b>

Table 1: Composition of Corpus Sets

### 3 Features

In order to create the suicide/non-suicide lyrics classifier, similar features to those used in the previous research were explored in conjunction with a set of original features. In all, there were 87 features that we explored.

#### 3.1 Vocabulary Features

While a few of our features were based on raw counts of types, tokens, and time of song (in seconds) alone, such as TTR (type-token ratio), they are mostly used to normalize many of the features in the following two sections. Below are the vocabulary features we explored:

- by type: type/token ratio (TTR) and type/time ratio
- by token: token/time ratio

#### 3.2 Syntactic Features

As in Le et al. (2011) and Stirman & Pennebaker (2001), we expected to find differences in the use of the passive construction and in the proportions of the first-person pronouns to the rest of the pronouns. We expected a greater use of passive constructions in the lyrics of the suicide lyricists in comparison to the non-suicide lyricists since it might signify a greater sense of disengagement from the external world. Additionally, we hypothesized a higher proportion of first-person pronouns to other pronouns in the suicidal lyrics since a common perception about suicide cases and depressive people in general is that they are more self-centered or that they are less concerned with others.

In addition to the exploration of the first-person pronouns and passive constructions, we also looked at the differences in the usage of mental-state verbs co-occurring with the first-person singular pronoun, including the use of verbs like *think* and *feel*. Our expectation here was that the suicide writers might use such constructions more often due perhaps to a preoccupation with thoughts and feelings and an inclination to think and feel more often than act.

These features were computed using the UAM corpus tool, which allows one to create autocoding rules and presents annotation statistics on the text level. Most of these count features were normalized by type, token, and time, but a few of them

consist of ratios between features, such as first-person singular pronouns to all other pronouns. Since the latter features were occasionally affected by data sparsity, we chose to deal with undefined values resulting from zeroes in the denominator by adding 0.01 to each count so that the resulting value would not be undefined.

#### 3.3 Semantic Class Features

Our expectations about the content of the suicide lyrics in comparison to the non-suicide lyrics was that they might deal with more negative, depressing subjects than positive ones. We also hypothesized that, as in Pennebaker & Stirman (2001), we would see a difference in the use of sexual words in the suicide lyrics (specifically, a heightened rate of sexual and death-related word usage). For these and other "semantic classes", we built word-lists consisting of terms relating to the target semantic classes.

The semantic classes considered were sensuality, action (specifically, verbs that signified some particular action), concreteness (specifically, nouns that represented concrete objects), death, love, depression, and drugs. We used the MPQA prior polarity word lexicon (Wiebe et al. 2005) to measure negative, positive, and neutral word usage. We counted the number of occurrences of these words in each song-text, normalizing the raw counts by type, token, and time. We also computed a number of features that consisted of ratios between raw counts, such as sensual words to positive words. Where applicable, we dealt with data sparsity using the same method described above, adding 0.01 to avoid undefined values.

Additionally, we used the AFINN (Nielsen, 2011) word-valence dictionary, which is a list of nearly 2,500 polar terms with associated polarity values (ranging from 5 to -5), to calculate the total and average polar value of each song. The total polar value was calculated by summing the polarity values for each polar term in a song-text while the average polar value was calculated by dividing the total polar value by the number of polar terms in a text.

#### 3.4 N-Gram Features

A Python script was written to build unique n-gram (for  $n = 1$  to  $n = 6$ ) profiles for the classes in the development set. (This set was used exclusively for this purpose and was not needed for any other features.) These unique n-gram profiles for

suicide and non-suicide lyricists were compared to the corresponding n-gram profiles of each song in the training and test sets to find out to what extent the n-grams in a given song overlapped with either n-gram profile.

In addition to percentage of overlapping n-grams, a number of features composed of the overlapping percentages were computed. The difference between overlapping n-gram percentages for each class was calculated for each value of n. For example, if a song's bigram profile overlapped with the non-suicide bigram profile at a rate of 6% and with the suicide bigram profile at a rate of 4%, the difference would be 2%. We also computed the average overlapping n-gram percentage for each class across all values of n. Finally, we calculated the difference between the average percentages of overlap for the non-suicide n-gram profiles and that for the suicide n-gram profiles.

## 4 Results

All features were input into Weka and a number of different ML algorithms were run to create a classifier. As a baseline for comparison, we used the majority-class prediction rate of 57.7%. The classifier was trained on the training set and then tested on the test set of different artists' songs. The most successful algorithm was the SimpleCart algorithm, which correctly classified the songs as either suicide or non-suicide 70.6% of the time and which achieved a 0.39 Cohen's kappa value. The correct classification rate represents an increase of 12.9% over the baseline. The SimpleCart algorithm achieved a precision, recall, and F-measure of 0.71, and an ROC Area value of 0.70. In Table 4 above, we report the confusion matrix for the test set.

While our classification statistics do not reach a satisfactory level, we believe that they indicate that we are on the right track and that this task can be tackled using NLP. Of the 87 features that were calculated, a number stood out as being most useful across numerous algorithms. Included among these features are the various n-gram features, the first-person singular + mental verb features, the concrete nouns, neutral terms, sensual words, and total polar value semantic class features, and the first-person singular and passive construction syntactic features.

a	b	<- classified as
29	17	a = suicide
15	48	b = non-suicide

Table 2: Confusion Matrix.

## 5 Discussion

The construction of a corpus for this type of task is beset by problems on all sides. Perhaps one of the largest issues is with the seemingly non-suicidal lyricists: whether these lyricists have passed away already or are still alive, there is no certainty that they would not have committed/will not commit suicide at some point after the point at which they are classified as non-suicidal. Perhaps one could try to use only those lyricists who died from non-suicidal causes late in life (say, after 60) since such lyricists might represent fairly safe cases, but even then there still would not be any certainty. Furthermore, there could be a situation in which a lyricist tried to commit suicide, but did succeed and news of the attempt was kept secret. Such lyricists might then be classified as non-suicides even though the amount of separation between them and the lyricists who were successful at committing suicide is next to nill. Although we acknowledge that this is a serious consideration, we believe that 1) it is a risk that we have to take in order to do this task since there seems to be no solution that guarantees 100% certainty and 2) the likelihood of committing suicide appears to be so low (even for artists) that it might not be such a bad course of action to assume that any given lyricist will not commit suicide unless he/she already has.

Though the vast majority of lyricists do not commit suicide, this fact leads directly to some of the other problems that afflicted the construction of our corpus. Since the number of lyricists who committed suicide is constrained, this leads to issues beyond the collection of as many songs as possible from such lyricists, which is a necessity. For example, the corpus of songs cannot simply be randomly split into sets because the we need to ensure that the songs we test on were not written by lyricists who composed songs that we trained on lest we merely learn the tendencies of those artists and not the abstract suicidal tendency that we are actually seeking. The same issue goes for the development set. However, even if we figured out a way to split the corpus into sets of the appropriate numbers of songs while taking into consideration



that the artists for each set must be unique, there are further factors that could skew the results. For example, we would ideally want the composition of each set of songs to be consistent from set to set in terms of the range of composition dates, the genres represented, etc. Although we attempted to take all of these factors into consideration in order to partition the data into sets, we realize that our method for doing so and the product of our labors leave much to be desired. In the future, we hope to work on refining our method so that it might optimize the partitioning of our corpus.

## 6 Further Research

Besides expanding the corpus to include many more non-suicide lyricists and (at the very least) to include more songs from each of the suicide artists, it would perhaps be fruitful to extend the analysis to other types of features and new lexicons since it has been demonstrated that this task could be solved using NLP.

## Acknowledgments

We would like to take this space to acknowledge again the use of the UAM corpus tool, which proved valuable to our analysis, the MPQA prior polarity lexicon, the AFINN word valence dictionary, the Weka machine learning suite, and the many online contributors of lyrics and word-lists. We would also like to thank Michael Flor for letting us use his own personal lemmatizer.

This material is based in part upon work supported by the National Science Foundation under Grant Numbers 0916280 and 1048406. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

- Justine Kao and Dan Jurafsky 2012. A Computational Analysis of Style, Affect, and Imagery in Contemporary Poetry. *NAACL-HLT* 2012, 8.
- Xuan Le, Ian Lancashire, Graeme Hirst, and Regina Jokel. 2011. Longitudinal detection of dementia through lexical and syntactic changes in writing: A case study of three British novelists. *Literary and linguistic computing*, 26(4):435-61.
- Y. Neuman, Y. Cohen, G. Kedma, and O. Nave. 2010. Using Web-Intelligence for Excavating the Emerging Meaning of Target-Concepts.

2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto, Aug-Sept 2010. IEEE Computer Society.

- F. Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *Proceedings of the ESWC2011 Workshop on Making Sense of Microposts: Big things come in small packages*, number 718 in CEUR Workshop Proceedings, Heraklion 2011.
- Mick O'Donnell 2008. Demonstration of the UAM CorpusTool for text and image annotation. *2010 Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Demo Session - HLT* Association for Computational Linguistics, 2008.
- J. P. Pestian, P. Matykiewicz, and M. Linn-Gust. 2012. Whats in a Note: Construction of a Suicide Note Corpus. *Biomedical Informatics Insights*, 2012:5 1-6.
- M. F. Porter 1980. An algorithm for suffix stripping. *Program*, 14(3):130137.
- S. Sohn, M. Torii, D. Li, K. Waghlikar, S. Wu, and H. Liu. 2012. A Hybrid Approach to Sentiment Sentence Classification in Suicide Notes. *Biomedical Informatics Insights*, 2012:5 (Suppl. 1) 43 50.
- S. W. Stirman and J. W. Pennebaker. 2001. Word Use in the Poetry of Suicidal and Nonsuicidal Poets. *Psychosomatic Medicine* 2001, 63:517-522.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* , 39(2-3):165210.
- Ian H. Witten and Eibe Frank. 1999. Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufman (1999).

# Unsupervised Word Class Induction for Under-resourced Languages: A Case Study on Indonesian

**Meladel Mistica**

The Australian National University  
meladel.mistica@gmail.com

**Jey Han Lau and Timothy Baldwin**

The University of Melbourne  
jeyhan.lau@gmail.com  
tb@ldwin.net

## Abstract

In this study we investigate how we can learn both: (a) syntactic classes that capture the range of predicate argument structures (PASs) of a word and the syntactic alternations it participates in, but ignore large semantic differences in the component words; and (b) syntactico-semantic classes that capture PAS and alternation properties, but are also semantically coherent (a la Levin classes).

We focus on Indonesian as our case study, a language that is spoken by more than 165 million speakers, but is nonetheless relatively under-resourced in terms of NLP. In particular, we focus on the syntactic variation that arises with the affixing of the Indonesian suffix *-kan*, which varies according to the kind of stem it attaches to.

## 1 Introduction

This research was motivated by the desire to semi-automatically develop a lexicon for a wide-coverage, precision grammar of Indonesian. Although these linguistically-motivated grammars are invaluable resources to the NLP community, the biggest drawback is the time required for the manual creation and curation of the lexicon. Our work aims to expedite this process by automatically assigning syntactic information to stems that make up the verbal elements, on the basis of predicting syntactico-semantic clusters based on distributional similarity.

However, one minor point becomes one major obstacle in this task: Indonesian is a relatively under-resourced language in terms of NLP. Therefore, many of the techniques that have been deemed successful in the inferring of syntactic information or inducing syntactico-semantic classes

are not available to us. Even studies that are considered lightweight minimally employ a part-of-speech (POS) tagger and chunker (Joanis et al., 2008), with many studies benefiting from the richness of the features that a syntactic parser provides (Schulte im Walde, 2006). In the case of Indonesian, there exist POS taggers (Pisceldo et al., 2009; Wicaksono and Purwarianti, 2010)<sup>1</sup> but no chunker or syntactic parser, and the reliance on such pre-processing tools is unrealistic.

We adhere to the notion that semantic similarity begets syntactic similarity as per Levin (1989), and so employ a distributional similarity method to learn our syntactic classes, based on a non-parametric Bayesian model. We experiment with learning both: (a) syntactic classes that capture the range of predicate argument structures (PASs) of a word and the syntactic alternations it participates in, but ignore large semantic differences in the component words; and (b) syntactico-semantic classes that capture PAS and alternation properties, but are also semantically coherent (a la Levin classes).

Here, we focus on the syntactic variation that arises with the affixing of the Indonesian suffix *-kan*. The specific morpho-syntactic behaviour of the *kan*-affixed verb is very much determined by the type of stem it attaches to, and its resulting behaviour varies from stem type to stem type (Kroeger, 2007; Vamarasi, 1999; Arka, 1993). The spectrum of variation induced by the affixing of *-kan* is not observed on all types of stems, and so being able to identify these superordinate types, representing the same morpho-syntactic variation, would assist greatly in accelerating lexicon development. It has been shown that Levin classes can be successfully induced employing unsupervised methods (Schulte im Walde, 2006; Kipper et al., 2006). We investigate the viability of automati-

<sup>1</sup>Although no POS tagger has been released for public use.

cally inducing coarser-grained types that represent morpho-syntactic variation, and we test whether the method we define is suited to such a task. Specifically, we focus on a case study detailed in Section 2 on the syntactic and semantic variation that arises with the affixing of the Indonesian suffix *-kan*. In Section 3 we outline our criteria in creating our gold standard data. Section 4 gives technical details of our methodology, and our interpretation of distributional similarity expressed in soft clusters derived using the hierarchical Dirichlet process (HDP). We present our results comparing our method employing HDP with a simpler benchmark system using hierarchical agglomerative clustering in Section 5, and also find that the method we employ in this study is better suited to discovering Levin-style classes rather than detecting morpho-syntactic variation, even though we had accommodated for syntactic structure in our model, by including functional words as structural indicators. We finally conclude with how we may extend this preliminary investigation.

Our contributions in this work are: (1) the demonstration that hierarchical Dirichlet processes are a highly effective way of modelling word similarity, outperforming simpler strategies; (2) the successful application of the syntax-semantic hypothesis of Levin to an under-resourced language based on distributional similarity analysis; (3) the finding that conflating semantic classes into superordinate types may be useful for annotating the lexicon, but when performing clustering tasks that employ distributional semantics, having a more semantically-oriented classification, such as Levin classes, are best suited for such methods, even when approximations are made to account for syntactic information; and (4) the demonstration that clustering based on semantic properties is a relatively strong predictor of deep syntactic lexical properties, and can be of great assistance in semi-automatically constructing a deep lexical resource for an under-resourced language

## 2 Background

Indonesian is an Austronesian language spoken by more than 165 million speakers in Indonesia (where it is the national language) and around the world (Gordon, 2005). Even with this status it still is an under-resourced language when it comes to NLP. For our case study, we aim to discover

groups of like stems that, when used predicatively in the same morphological context, give rise to the same syntactic behaviour. That is, we aim to induce classes of stems that exhibit the same syntactico-semantic behaviour when they have the same morphological marking.

Predictions on syntactico-semantic properties of stems via morphological processes have also been explored for English (Grimshaw, 1990). Although Grimshaw's account of nominalisation restrictions with the English suffix *-ing* can be explained with a more general theory of argument structure, she also shows that the nominalisation of certain predicates in this way exclude certain lexical classes, namely psychological predicates as shown in Example (1).

- (1) a. \*The (movie's) depressing**ing** of the audience.
- b. \*The worry**ing** of the public.

The morpho-syntactic study presented in this paper is specific to Indonesian, but these lexical changes initiated by morphological processes can be a source of investigation into syntactico-semantic properties of lexemes for a variety of languages including English. For our case study we look into the Indonesian suffix *-kan*, which is generally described as a morpheme that triggers a lexical rule that increases valency. It can introduce a benefactive object, form a causative construction, or apply other semantic changes. Examples (2) and (3) show the benefactive, and causative uses, respectively:

- (2) a. *Dia membeli buku itu untuk Mary.*  
s/he AV+buy book this for M  
“(S)he bought a book for Mary.”
- b. *Dia membelikan Mary buku itu.*  
s/he AV+buy+KAN M book this  
“(S)he bought Mary a book.”
- (3) a. *Orang-orang mengungsi.*  
person-person AV+take-refuge  
“The people took refuge.”
- b. *PBB mengungsikan orang-orang.*  
U.N. AV+refuge+KAN person-person  
“The U.N. evacuated the people.”

In the second line of each of these glossed examples, AV stands for *actor voice*, which means that the verb is active. This is marked by the prefix *me-* plus a homorganic nasal, which can be realised

as  $m$ ,  $n(g|y)$  or  $\emptyset$ . This verb behaves in a similar fashion to English verbs in an active sentence. We limit the examination of verbs in this study to those that exhibit the actor voice (AV) marking.

Linguists have tried to characterise stems according to their behaviour when affixed with *-kan* (Dardjowidjojo, 1971; Arka, 1993; Vamarasi, 1999). In particular, Vamarasi (1999) claims that *kan* is a good diagnostic for separating unaccusative from unergative stems, which predicts their morphosyntactic behaviour. However the facts of *-kan* seem more intricate than this characterisation. Even though the causative and benefactive constructions uses of *kan* are the most commonly cited, its usage is much more varied and nuanced, as shown by Kroeger (2007), which is why we chose this morpho-syntactic construction as our case study.

Since the early '90s, the tools and resources employed in valency acquisition tasks have become increasingly sophisticated and linguistically-rich. One of the earlier examples of this is by Brent (1993), who employs a system based on deterministic morphological cues to identify predefined syntactic patterns from the Brown Corpus. Manning (1993) employs a shallow parser or chunker in order to acquire subcategorisation frames from the New York Times. Schulte im Walde (2002) induces subcategorisation information for German with the use of a lexicalised probabilistic context free grammar (PCFG), and O'Donovan et al. (2005) employ the richly-annotated Penn Treebank in achieving this endeavour. In terms of resources, our work most closely resembles Brent (1993), in that we rely mainly on linguistic knowledge based on simple lexical features. However, the way linguistic knowledge is learned and applied is quite different, as we will see in Section 3

In terms of the methodology, the studies that we look to are those systems that are built to disambiguate and/or discover syntactico-semantic Levin-style classes, rather than systems that aim to induce valency or syntactic frame information from corpora. These can be built in a supervised fashion as in Lapata and Brew (2004) or tackled as a clustering task as in Schulte im Walde (2006) or Bonial et al. (2011). Lapata and Brew (2004) develop a semi-supervised system that generates, for a given verb and its syntactic frame, a probability distribution over the Levin verb classes. They then use this system to disambiguate tokens using collocation information. Our system, like Schulte

im Walde (2006), uses an unsupervised clustering approach. In her approach, Schulte Im Walde employs hierarchical agglomerative clustering over parse features to discover word classes in German, and evaluates using manually-created gold-standard data.

### 3 Evaluation Data

This section describes how we arrive at the two evaluation sets we use in our experiments.

#### 3.1 Forming Levin Classes

We use VerbNet 3.2<sup>2</sup> as our guide for forming Levin classes for Indonesian, and rely on their translation to determine membership for the class, for a particular sense of that verb.

We have 30 stems that we group into 16 Levin classes. Unlike the types we form in Section 3.2, which have unique membership, a lexical item can appear in multiple classes as appropriate. For example *baca* “read” has membership in both VerbNet classes **say-27.7** and **learn-14**. We show a subset of Indonesian Levin classes we develop based on VerbNet 2.3 in Table 1.

#### 3.2 Forming Superordinate Levin Types

These superordinate types combine Levin classes to form groups of stems that behave in the same way syntactically, but may not all be synonyms of each other. In determining the coarse-grained superordinate types, we did not simply want to group stems according to intuition. Rather, we were after an explicit description of the syntax and semantics of grouped stems that all behave in the same way when affixed with *-kan*. Stems that are grouped together should exhibit the same semantic shifts. That is, if affixing *-kan* to a stem gives rise to a causative meaning, then its corresponding group member will also produce a causative meaning when *-kan* is applied to the stem. Also, if adding a *-kan* does not increase the valency for a stem in a particular group, then its corresponding group member will also exhibit the same syntactic behaviour.

In order to achieve this, we map out the different behaviour of verb stems when they occur in the morphological patterns (a) and (b):<sup>3</sup>

<sup>2</sup><http://verbs.colorado.edu/~mpalmer/projects/verbnet/downloads.html>

<sup>3</sup>As mentioned earlier in Section 2, AV stands for *actor voice*, and can be likened to an English verb in an active sentence.

Indonesian Members	VerbNet Class
<i>beri</i> “give” <i>jaja</i> “hawk/sell”, <i>pinjam</i> “lend”	give-13.1
<i>kenang</i> “think” <i>kenal</i> “know” <i>ingat</i> “remember”	consider-29.9
<i>mati</i> “die”, <i>tewas</i> “perish”	disappearance-48.2
<i>susup</i> “duck down”, <i>singkir</i> “get out of way”	avoid-52
<i>timpa</i> “hit” <i>hantam</i> “hit/blow” <i>tabrak</i> “hit”	hit-18.1
<i>baca</i> “read” <i>tulis</i> “write”	say-37.7
<i>baca</i> “read” <i>hafal</i> “memorize”	learn-14

Table 1: Subset of the mapping of Levin classes into Indonesian

- (a) ME  $N$ +stem  
AV+stem
- (b) ME  $N$ +stem+KAN  
AV+stem+KAN

We map out the variation of arguments for pattern (a) with only the AV prefix, i.e. ME  $N$ +stem, and then note the changes when the stem has both the actor AV and *-kan* affixes, i.e. pattern (b) ME  $N$ +stem+KAN. We also track the semantic changes relative to the stem for these two patterns and found that 25 verb stems found their way into 8 verb types.<sup>4</sup> This formed one of our evaluation sets in our experiments (see Mistica (2013) for further details on forming these superordinate types).

In the interests of space, we only present two out of the 8 manually-induced verb types in Table 2. Below each of the types, we show the syntactic and semantic changes that determine our verb types or subclasses.

## 4 Method

We define our features in terms of the context of occurrence of our target lexeme, and employ hierarchical agglomerative clustering (HAC) over these features in two ways: (1) directly over the raw word frequencies; and (2) over extracted semantic features learned via the contexts of occurrence, which are represented as topic probabilities.

We use Indonesian Wikipedia<sup>5</sup> as our text collection, and remove mark-up with Wikiprep,<sup>6</sup> then tokenise with the English-trained models of OpenNLP.<sup>7</sup> The total word count of the text collection is approximately 26 million words. In the

<sup>4</sup>We had also manually grouped stems from other word classes: 48 noun stems were grouped into 13 subclasses; and 27 adjective stems were grouped into 5, giving us a total of 100 stems with the 25 verbs, but we only report on the verb experiments.

<sup>5</sup><http://dumps.wikimedia.org/idwiki/>

<sup>6</sup><http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep>

<sup>7</sup><http://opennlp.apache.org/> Our experiments showed that OpenNLP’s English models performed better than a rule-based Malay sentence tokeniser (Baldwin and Awab, 2006).

next section we summarise the features we use in our experiments, in addition to outlining our clustering method.

### 4.1 Feature Engineering

Our features determine how we collect unigrams from the text collection. We collect these unigram features from 735 lexemes that we were able to identify as possible *-kan* hosts. These 735 lexemes had stems that belonged to any of the open class categories in Indonesian (noun, adjective or verb).

In our preparation of the Wikipedia data, we include function words as a means to infer structural information. Because we do not use a parser to explicitly obtain syntactic features, this is how we approximate this kind of information.

We use three main feature types in our task: (1) **morph**  $\in$  ‘k’, ‘mk’, ‘smk’; (2) **win**  $\in$  1 to 5; and (3) **context**  $\in$  ‘+’ (forward), ‘-’ (backward).

**Morphological features (morph):** These are contextual features for different morphological forms of the target lexeme, where: ‘s’ stands for *stem*, i.e. the unaffixed lexeme; ‘m’ stands for the AV variant of the lexeme, based on pattern (a) from Section 3.2; and ‘k’ stands for the KAN suffixed form of the AV variant of the lexeme, based on pattern (b) from Section 3.2. An example of the ‘s’, ‘m’ and ‘k’ variants of *beli* “buy” are *beli*, *membeli*, and *membelikan*, respectively. These morphological features determine whether the unigram features we collect for a lexeme are based on instances of ((s)m)k forms found in the text. We experiment with the context features based on these morphological variants in isolation and also in combination. For example, ‘mk’ would capture context features for the *membeli* and *membelikan* variants of the stem *beli* “buy”.

**Window Size (win):** This stipulates the context window size, relative to individual occurrences of the target lexeme, and can take a value of 1–5.

<b>Example Type A:</b> <i>acuh</i> “to heed”, <i>terjemah</i> “translate”, <i>mandi</i> “bathe”		
MEN+V <sub>1</sub>	–	–
MEN+V <sub>1</sub> +KAN	<NP <sub>a</sub> , NP <sub>b</sub> >	DO <sub>to</sub> ( [NP <sub>a</sub> ], [V <sub>1</sub> TO( [NP] ) ] )
<b>Example Type B:</b> <i>dengar</i> “hear”, <i>kenang</i> “think of”		
MEN+V <sub>3</sub>	<NP <sub>a</sub> , NP <sub>b</sub> >	HAPPEN <sub>to</sub> ( [NP <sub>b</sub> ], [ V <sub>3</sub> TO([NP <sub>a</sub> ] ) ] )
MEN+V <sub>3</sub> +KAN	<NP <sub>a</sub> , NP <sub>b</sub> >	DO <sub>to</sub> ( [NP <sub>a</sub> ], [ V <sub>3</sub> TO( [NP <sub>b</sub> ] ) ] )

Table 2: Manually generated verb Types (‘–’ = no attested word form in the text; ‘{...}’ = optional)

**Context Features (context):** We look at backward (‘–’) or forward (‘+’) context unigrams.

## 4.2 Clustering Stems

We employ hierarchical agglomerative clustering (HAC) in two ways: (1) over the raw frequencies of words based on the feature representations defined in Section 4.1; and (2) over the output of the distributional semantic modelling (HDP) discussed in Section 4.3. The output of this step produces topic models. In other words, we perform HAC over raw unigram frequencies and induced topic models from these raw frequencies to ascertain the usefulness of the HDP step.

To compute the distance between a pair of patterns, we use Squared Euclidean, and for the linkage criterion for merging clusters we use weighted linkage clustering (WPGMA). We compare the output of HAC with the flat-structured gold-standard classes. In order to induce flat clusters from the hierarchical output of HAC, we apply a similarity threshold  $t = 0.825$  to determine which instances should be grouped together.

## 4.3 Modelling Distributional Similarity

Distributional semantic models are commonly employed in the induction and disambiguation of word senses (McCarthy and Carroll, 2003; Lapata and Brew, 2004; Brody and Lapata, 2009; Lau et al., 2012), and to a lesser extent, in learning syntactic classes and diathesis alternation behaviour (Parisien and Stevenson, 2011; Bonial et al., 2011). We infer lexical similarity and soft word clusters using topic modelling, based on a hierarchical Dirichlet process (HDP: Teh et al. (2006)), a non-parametric extension of latent Dirichlet allocation (LDA: Blei et al. (2003)). LDA is a Bayesian generative topic model that learns *latent* topics for a collection of documents

based on the *observable* words. Our definition of a document is a target lexeme and the observable words that surround the target lexeme (based on the window size in the parameter settings).

Formally, in LDA a topic is associated with a multinomial distribution of words, and each document (i.e. lexeme) in the collection is associated with a multinomial distribution of topics. HDP relaxes the constraint in LDA where the number of topics  $T$  is fixed, and learns  $T$  based on the training data using Dirichlet processes (DPs).

## 4.4 Evaluation

We develop two baseline systems to compare our results against: (1) majority class; and (2) random class assignment based on a uniform class distribution. The random scores reported are based on the median of 11 random assignments.

We use pairwise precision ( $pP$ ), recall ( $pR$ ), and F-score ( $pF_1$ ) to evaluate our generated clusters, relative to the gold-standard word classes, as described by Schulte im Walde (2006).

## 5 Results

We perform two experiments. First, we apply the hierarchical Dirichlet process (HDP) to produce topic probabilities, over which we perform HAC. Second, we perform HAC over the raw unigram features (NoHDP), as our benchmark system, a method also employed by systems such as Schulte im Walde (2002) for German and Jurgens and Stevens (2010) for English word sense induction. In both cases, we base our experiments on the 735 lexemes identified as being able to be affixed with *-kan*, and the unigram features from Section 4.1. Note, however, that evaluation is based on the subset of the 735 lexemes which were manually classified into classes and types in Section 3.

We employ a *bagging* approach (sampling with

System	Maj.	Rand.	ON-ALL	ON-VERBS
LEVIN-HDP			<b>.174</b>	<b>.367</b>
LEVIN-NOHDP	.114	.065	.057	.111
TYPES-HDP			<b>.281</b>	.261
TYPES-NOHDP	.271	.140	.026	.152

Table 3:  $pF_1$  score comparing benchmark system NOHDP with our HDP system for Levin Classes (LEVIN) and our coarser-grained TYPES

A	<i>main</i> “play”, <i>nyanyi</i> “sing”, <i>gesek</i> “scrape”
B	<i>irim</i> “send”, <i>hantar</i> “place”
C	<i>dapat</i> “get”, <i>menang</i> “win”, <i>terima</i> “receive”

Table 4: Induced groups with no known categorised words

replacement) to ascertain the best parameters to apply to our 735 lexemes in terms of the unigram features we define in Section 4.1.

Given the discovered parameters, we report our results in Table 3. The label ON-ALL for all HDP systems are systems that have had topics induced from all 735 stems (made up of not only verbs, but also nouns and adjectives), while ON-VERBS only induces topics from a subset of the 735 lexemes whose stems are also verbs, even though we only evaluate on verbs in these experiments.

We observe in Table 3 that HDP consistently outperforms NO-HDP systems. Furthermore, the LEVIN-HDP system outperforms the Random (“Rand.”) and the Majority Class (“Maj.”) baselines, as well as the benchmark NOHDP system. The TYPES-HDP system, on the other hand, barely exceeds the Majority Class baseline with the ON-ALL experiment, and fails to do so with the ON-VERBS experiment.

## 6 Discussion

For our error analysis, we examine a sample of the resulting stem groups from the Levin Class experiments. Table 4 shows membership of all stems found in four separate clusters. The lexemes from these particular groups do not have membership into any of the gold standard Levin classes, unlike the groups formed in Table 5. In this table, the top half are groups that match our Levin classes, part of which is presented in Table 1, and the bottom half are groups that do not match Levin classes.

Group A from Table 4 has 3 verbs — *main*

D	<i>singkir</i> “get out of way”, <i>susup</i> “duck down”
E	<i>baca</i> “read” <i>hafal</i> “memorise”
F	<i>terjemah</i> “translate” <i>tulis</i> “write”, <i>muat</i> “insert/contain”
G	<i>paksa</i> “force” <i>pinjam</i> “lend” <i>hapus</i> “wipe off/vanish/blot out”

Table 5: Induced groups with known categorised words

“play”, *nyanyi* “sing”, and *gesek* “scrape” — which may initially seem not to form a semantically coherent group, however they all are associated with producing music: *main* “play” is used to describe the playing of most musical instruments, and *gesek* “scrape/rub” is used for string instruments, such as violins or cellos. Group B has members that describe movement from one place to another, as does Group C.

Groups D and E in Table 5 faithfully replicate the Levin Classes **avoid-52**, and **learn-14** from Table 1. However, Groups F and G seem to not form coherent semantic groups.

## 7 Conclusion

We have explored the question of whether distributional similarity models can be used to learn deep syntactic features for an under-resourced language, namely Indonesian. Our results demonstrate that hierarchical Dirichlet processes are a highly effective way of modelling word similarity, and outperform a simpler strategy of simply applying HAC over raw frequencies. We have also shown that learning classes geared toward the potential morpho-syntactic alternations of stems, while conflating the semantics of the stem are too coarse for this particular method. The experiments that used true Levin classes to evaluate against performed much better in comparison to the baselines, than did the experiments where we induced our manually constructed coarse-grained types. Although resources and tools are limited for Indonesian NLP, we would need to model syntactic structure more effectively to gain success in predicting lexical types rather than Levin classes.

## References

- I Wayan Arka. 1993. Morphological aspects of the -kan causative in Indonesian. Master's thesis, The University of Sydney, Sydney, Australia, November.
- Timothy Baldwin and Suád Awab. 2006. Open source corpus analysis tools for Malay. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 2212–5, Genoa, Italy.
- David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Claire Bonial, Susan Windisch Brown, Jena D. Hwang, Christopher Parisien, Martha Palmer, and Suzanne Stevenson. 2011. Incorporating coercive constructions into a verb lexicon. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, pages 72–80, Portland, USA.
- Michael R. Brent. 1993. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19(2):243–262.
- Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the EACL (EACL 2009)*, pages 103–111, Athens, Greece.
- Soenjono Dardjowidjojo. 1971. The meN-, meN-kan, and meN-i verbs in Indonesian. *Philippine Journal of Linguistics*, 2:71–84.
- Raymond Gordon. 2005. *Ethnologue: Languages of the World*. SIL International, Dallas, USA.
- Jane Grimshaw. 1990. *Argument Structure*. The MIT Press, Cambridge, USA.
- Eric Joanis, Suzanne Stevenson, and David James. 2008. A general feature space for automatic verb classification. *Natural Language Engineering*, 14(3):337–367.
- David Jurgens and Keith Stevens. 2010. HERMIT: Flexible clustering for the SemEval-2 WSI task. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 359–362, Uppsala, Sweden.
- Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending VerbNet with novel verb classes. In *Proceedings of LREC 2006*, pages 1027–1032, Genoa, Italy.
- Paul R. Kroeger. 2007. Morphosyntactic vs. morphosemantic functions of Indonesian '-kan. In Joan Bresnan, Annie Zaenen, Jane Simpson, Tracy Holloway King, Jane Grimshaw, Joan Maling, and Christopher D. Manning, editors, *Architectures, rules, and preferences: variations on themes*, CSLI Lecture Notes, pages 229–251. CSLI Publications.
- Mirella Lapata and Chris Brew. 2004. Verb class disambiguation using informative priors. *Computational Linguistics*, 30(1):45–73.
- Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2012)*, pages 591–601, Avignon, France.
- Beth Levin. 1989. *English Verb Classes and Alternations: A preliminary investigation*. The University of Chicago Press, Chicago, USA.
- Christopher D. Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceeding of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 235–242, Columbus, USA.
- Diana McCarthy and John Carroll. 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.
- Meladel Mistica. 2013. *An Investigation into Deviant Morphology: Issues in the Implementation of a Deep Grammar for Indonesian*. Ph.D. thesis, The Australian National University, Canberra, Australia.
- Ruth O'Donovan, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 2005. Large-scale induction and evaluation of lexical resources from the Penn-II and Penn-III treebanks. *Computational Linguistics*, 31(3):229–365.
- Chris Parisien and Suzanne Stevenson. 2011. Generalizing between form and meaning using learned verb classes. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, Boston, USA.
- Femphy Pisceldo, Ruli Manurung, and Mirna Adriani. 2009. Probabilistic part-of-speech tagging for Bahasa Indonesia. In *Proceedings of the Third International MALINDO Workshop*, Singapore.
- Sabine Schulte im Walde. 2002. A subcategorisation lexicon for German verbs induced from a lexicalised PCFG. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, volume IV, pages 1351–1357, Las Palmas de Gran Canaria, Spain.
- Sabine Schulte im Walde. 2006. Experiments on the automatic induction of German semantic verb classes. *Computational Linguistics*, 32(2):159–194.
- Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.
- Marit Kana Vamarasi. 1999. *Grammatical relations in Bahasa Indonesia*, volume 93 of *Series D*. Pacific Linguistics, Canberra, Australia.
- Alfan Farizki Wicaksono and Ayu Purwarianti. 2010. HMM based part-of-speech tagger for Bahasa Indonesia. In *Proceedings of the 4th International MALINDO Workshop (MALINDO2010)*, Depok, Indonesia.



# An Efficient Active Learning Framework for New Relation Types

**Lisheng Fu**

Computer Science Department  
New York University  
lf1099@nyu.edu

**Ralph Grishman**

Computer Science Department  
New York University  
grishman@cs.nyu.edu

## Abstract

Supervised training of models for semantic relation extraction has yielded good performance, but at substantial cost for the annotation of large training corpora. Active learning strategies can greatly reduce this annotation cost. We present an efficient active learning framework that starts from a better balance between positive and negative samples, and boosts training efficiency by interleaving self-training and co-testing. We also studied the reduction of annotation cost by enforcing argument type constraints. Experiments show a substantial speed-up by comparison to the previous state-of-the-art pure co-testing active learning framework. We obtain reasonable performance with only 150 labels for individual ACE 2004 relation types.

## 1 Introduction

Relation extraction aims to discover the semantic relationship, if any, between a pair of entities in text. This structured information can be used to build higher-level applications such as question answering and other text mining applications.

Relation extraction was intensively studied as part of the multi-site ACE [Automatic Content Extraction] evaluations conducted in 2003, 2004, and 2005. For 2004, six major relation types were defined. Each relation mention takes two entity mention arguments in the same sentence. In annotating text, each entity mention pair within one sentence will be labeled if it involves one of the relation types. As part of ACE, substantial hand-annotated corpora marked with entities and relations were produced. For example, the ACE 2004 corpus had in total about 5,000 relation instances (and about 45,000 same-sentence entity pairs not bearing one of these relations). These large training corpora stimulated research on the supervised training of relation extractors, with

considerable success: the best systems, when given hand-tagged entities, correctly identify and classify relations with an F score above 70% (Jiang and Zhai 2007).

Although supervised methods were effective, annotating a corpus of this size is too expensive in practice to serve as a model for developing new extractors: it requires consideration of 50K instances, of which only a small portion involve the target relation type. In consequence, most research has focused on reducing the annotation cost through semi-supervised learning methods such as bootstrapping systems. However, with limited labeled data, those semi-supervised systems failed to come close to the supervised level of performance. Their performance also varies with the distribution of seeds.

Recent studies have proposed new ways of reducing the annotation cost by using active learning. The advantage of active learning is that it can achieve reasonable performance, and even performance comparable to the supervised version, with few labeled examples, due to its ability to selectively sample unlabeled data for annotation.

To further reduce the annotation cost and provide an efficient framework for rapidly developing relation extraction models, we combine active learning with semi-supervised methods, provide solutions to the imbalanced seed set and uneven co-testing classifiers, and optionally incorporate argument type constraints. Most relation types now achieve reasonable performance with only 150 labeled instances. Section 2 gives more related work in detail. Section 3 describes the enhancements we have made. Section 4 reports the experimental results and the improvement in performance when only a few instances have been labeled. Section 5 concludes the paper.

## 2 Related Work

For reducing the cost of annotation in the task of relation extraction, most prior work used semi-

supervised learning. (Uszkoreit 2011) introduced a bootstrapping system for relation extraction rules, which achieved good performance under some circumstances. However, most previous semi-supervised methods have large performance gaps from supervised systems, and their performance depends on the choice of seeds (Vyas et al., 2009; Kozareva and Hovy, 2010).

Recent studies have shown the effectiveness of active learning for this task. (Zhang et al., 2012) proposed a unified framework for biomedical relation extraction. They used an SVM as the local classifier and tried both uncertainty-based and density-based query functions and showed comparable results for the two methods. They also proposed using cosine-distance to ensure the diversity of queries.

(Roth and Small 2008) used a dual strategy active learner (Donmez, Carbonell, & Bennett 2007) in their pipeline models of segmentation, entity classification and relation classification at the same time. They also adopted a regularized version of the structured perceptron (Collins 2002) instead of SVM and reported better results in active learning. Their work simulated the whole pipeline in active learning to achieve relation extraction, but had no specific research on the stage of relation extraction in the pipeline.

(Zhang 2010) proposed multi-task active learning with output constraints as a generalization of multi-view learning. The multi-task method relied on constraints on output between different tasks; this might be extended to situations where we need to learn relation sub-types as well as types, but was not applicable when relation extraction is an individual task.

Multi-view learning in a co-testing framework was used in (Sun and Grishman 2012). This paper proposed an LGCo-testing framework in which the local view is a maximum-entropy model with local features, and the global view is based on the distributional similarity in a large unlabeled corpus of the phrases between the two entity mentions of a relation. Extractor training was faster than with alternative active learning methods – much faster than with sequential annotation.

There has been research on combining different learning methods with active learning to obtain further improvement. (Song et al. 2011) used variants of SVM to apply semi-supervised learning after active learning in protein-protein interaction extraction.

The current paper adopts the earlier co-testing framework (Sun and Grishman 2012) and exam-

ines some of the design issues in order to achieve substantial further speed-ups.

### 3 Method

#### 3.1 Framework

In active learning, users are asked to judge whether a particular sentence expresses the target relation between two entity mentions. For a fixed number of queries (fixed annotation cost), active learning aims to achieve the highest performance possible. The work described here builds on a state-of-the-art co-testing based active learning algorithm (Sun and Grishman 2012). Our framework starts with a better initial setting (section 3.2), and then interleaves self-training with querying (section 3.3). We adjust for imbalanced classifiers (section 3.4) to improve query selection. By enforcing entity type constraints (section 3.5), the annotation cost could be further reduced. This framework is able to build a bridge between labeled data and unlabeled data more rapidly than previous pure co-testing based active learning.

The overall procedure is as follows:

```

Let:
U: unlabeled data
V: labeled data
(Labeled positive [relation] or negative [non-relation])
L: Local classifier
G: Global classifier

BEGIN
  // Initial set, section 3.2
  V = seed set
  Add Non-relations to V [see text]
  Train L, G on V
  REPEAT
    //Co-testing based on L and G, section 3.3
    P = {x ∈ U | G(x) = pos & L(x) = neg}
    N = {x ∈ U | G(x) = neg & L(x) = pos}
    Q = 5 queries selected from P ∪ N, preferring P;
    FOR each q ∈ Q
      //Entity type rules, section 3.5
      IF q violates entity type constraints
        THEN V += <q, neg>
        ELSE V += <q, user-assigned label>
    END IF
  END FOR
  Retrain L, G on V
  //Interleaved self-training, section 3.4
  Self-Train using both L, G

```

```
    to obtain positives and negatives and add to V
  Retrain L, G on V
END REPEAT
END
```

### 3.2 Non Relation Approximation

To initiate active learning, we require a small number of seeds (5 in our experiments) for the target relation type. To train the initial model, we also need negative samples. If a small set of negative samples were sufficient, we could ask the user to provide them. However, a small negative set would not be representative of the entire data space, which has far more negative instances than positive ones.<sup>1</sup> As a result, such an initial model gives poor performance; queries in early iterations appear irrelevant to the target relation. Better approximating the negative background by adding a certain number of high-confidence negative samples automatically will give the model the ability to distinguish most negative samples from the very beginning, thus accelerating initial learning.

Random sampling could be used to obtain the negative examples because of the sparsity of positives. However, there is the risk that random sampling may introduce too many false negatives, which is not acceptable for the initial set, even though active learning can deal with a certain degree of noise. To overcome this problem, we train an initial model by incrementally adding more probable non-relations. Since every relation is defined under entity type constraints, we have a subset of the unlabeled data in which the mention pair violates these constraints on the target relation. The instances in this subset are strongly assured not to be target relations if the entity types are hand-labeled, and somewhat more weakly assured if labeled by a NE tagger. By sampling from this subset of non-relations, we safely approximate the non-relation background of the unlabeled data and foster the early learning of the entity type rules. Thus the queries will also be more meaningful to users even at the beginning of the active learning process.

In implementing the sampling, we use the metric of how much of the non-relation subset we have learned instead of specifying a fixed number of instances. We train the model (a basic local feature classifier, the same as that in co-

testing, section 3.3) on the labeled instances, apply the classifier to the so-far-unlabeled instances of this subset, and rank the instances by their uncertainty. We repeatedly select the five most uncertain instances, add them to the labeled set, and retrain the model until the model gives mostly correct predictions on classifying the non-relations in this subset. In the experiments, it is tuned to be 99% accurate on non-relations when the model has roughly balanced precision versus recall on target relations. The balanced model will be a better initial model for later active learning. Meanwhile, the way we add non-relations also enforces early learning of entity type constraints.

### 3.3 Co-testing based query selection

When the initial set is ready, we can start selective sampling and pose queries to improve the model. We use a co-testing method similar to LGCo-Testing (Sun and Grishman 2012), the state-of-the-art active learning algorithm for relation type extension, but give preference to the weaker classifier to get some additional benefit in the early iterations.

LGCo-Testing uses co-testing based on the local view and the global view to select queries. The local classifier uses a rich set of lexical and syntactic features (from both constituent and dependency parses) as well as semantic type information for the arguments. (Zhou et al. 2005; Jiang and Zhai 2007) studied the effectiveness of different features. The global classifier relies on the similarity of relation phrases (the words between the entity mentions), computed based on the shared contexts of these phrases across a large news corpus. The global classifier returns the relation type of the labeled instances to which the unlabeled instance is most similar (a  $k$ -nearest-neighbor strategy, with  $k=3$ ).<sup>2</sup> The instances on which the two classifiers disagree is the contention set, from which queries are selected. Elements of the contention set are ranked by the KL-divergence, and elements with greater divergence are preferred as queries (because they are likely to be more informative in updating one of the models). Because of the additional knowledge from the global view, this method outperforms other methods in active learning for relation extraction, and thus we choose this method as our query selection function.

---

<sup>1</sup> The number of non-relation instances (mention pairs that are not the target type) is usually much larger than the number of target relations. In ACE 2004, it's about 25 times larger than the most frequent relation, EMP-ORG.

---

<sup>2</sup> We closely followed the classifier design in (Sun and Grishman 2012) so that our results would be comparable; the reader is referred there for more details.

While the global view provides valuable additional knowledge, the global classifier, in practice, gives few positive predictions. In principle, when the two classifiers are evenly matched, co-testing should work quite well at selecting informative instances. In this case, their settings favor instances with a positive prediction from the local classifier and a negative prediction from the global classifier, thus influencing the selection of queries. However, in terms of diversity of queries, the global classifier is more capable of discovering unseen instances in the local feature space.

Active learning systems that are based on co-testing may have a similar problem. So we tried to compensate for this by giving preference to the global classifier. In the contention set, the system will first pick as queries instances that the global classifier believes to be positive, and then pick instances that the local classifier predicts to be positive (this may result in selecting queries only from the global classifier in one iteration). The contention set works based on uncertainty. Giving priority to the global classifier is similar to the preference for density in active learning, which usually works better at few labels (Donmez, Carbonell, & Bennett 2007). To save computing time, the selection is only made from the top entropy instances (1000 in our experiments). When there is a substantial amount of annotated data, the local feature model will be able to cover the diversity from the global view. At this point, the contention set will only have examples that the local classifier predicts positive among the top entropy instances, and the priority to the global classifier will not make changes to query selection. We naturally transition to the original uncertainty-based co-testing. This actually gives a kind of mixture of uncertainty-based and density-based methods, which is expected to give better overall performance.

### 3.4 Interleaving Self-training

At each iteration of co-testing, the contention set from the local and global classifiers will be the candidate set for queries to be given to the user (section 3.3). We would also like to make use of the agreement set – the elements on which the classifiers agree – to further improve the model. We can do so by applying a semi-supervised method, akin to bootstrapping. To integrate this with active learning, we propose to automatically label selected elements of the agreement set at

each iteration, thus extending the knowledge directly provided by the user.

We employed the same models as those in active learning for estimating the confidence. In this task, positives are sparse, while negatives are frequent, so we distinguish the strategies for bootstrapping the two classes in the agreement set. For positives in the agreement set, we set a threshold on the local classifier to select sufficiently confident instances in order to avoid errors even when the model is small. We picked the threshold (0.8) based on our observation of early iteration self-training results. The global classifier works as a constraint to avoid semantic drift. (Sun and Grishman 2011) showed that clusters in the global view could be effective constraints in semi-supervised relation extraction. The global classifier, based on the similarity to the few labeled instances, provides a much stricter constraint on predicting an instance to be positive, so no threshold was required. Among those positive agreement instances satisfying the local classifier threshold, we select the most confidently classified instances to label.

In using those instances which both classifiers agree to be negative, we tend to be greedy. In fact, this is again selecting non-relations from unlabeled data, just as in the initial set setting. In the middle of the active learning, the model is more robust to noisy data, and this negative agreement set is also closer to a pure non-relation set. We employ random sampling on this set to emphasize the diversity since we are less concerned about accuracy. To maintain the balance of positives and negatives in the model, we let self-training produce the same number of positives and negatives. To avoid semantic drift away from human annotation, for each class (positive and negative), we limit the number of self-trained instances to be the same as the number of queries (5) at each iteration.

### 3.5 Entity Type Constraints

Relations are defined within entity type constraints. For instance, the EMP-ORG relation is limited to the types (PER – ORG), (PER – GPE), (ORG – ORG), (ORG – GPE), and (GPE – ORG) in ACE 2004.<sup>3</sup> In supervised learning, this is usually not a big problem.

---

<sup>3</sup> PER = person, ORG = organization, GPE = geo-political entity: a location with a government.

	30 iterations		stopping point: iterations	at stopping point		supervised learning
	baseline	our system		baseline	our system	
EMP-ORG	58.13	71.52	200	76.81	76.66	75.63
PHYS	34.63	41.16	200	57.85	64.71	67.39
GPE_AFF	18.18	43.01	119	53.69	53.68	63.33
PER-SOC	74.29	68.87	47	65.67	73.13	73.28
ART	25.93	43.33	31	25.45	43.33	74.36
OTHER-AFF	16.67	50.00	22	10.26	50.00	52.17
Overall	37.97	52.98	103	48.29	60.25	67.69

Table 1. Comparison with baseline (F1 score). The overall F score is the direct average of 6 types

Type	# queries in total	#queries that filters apply	Ratio
EMP-ORG	1000	91	9.1%
PHYS	1000	106	10.6%
GPE-AFF	590	84	14.2%
PER-SOC	234	64	27.3%
ART	151	54	35.8%
OTHER-AFF	105	56	53.3%

Table 2. Instances auto-labeled by type constraints

When the number of instances is large enough, the statistical model will effectively incorporate these entity type constraints as long as entity types are extracted as features. However, in active learning, even with suitable training examples, we will select and present to the user some instances violating these constraints. Applying explicit type filters would save a certain amount of human labeling effort. In practice, this still depends on the quality of the NE tagger. In the experiment section, we show that we can save a certain amount of annotation by using these simple constraints on hand-annotated entities. Since the savings is substantial, especially on some sparse types, it will be still helpful when using an imperfect NE tagger. A similar rule can be constructed to reject candidate relations where the two arguments are co-referential.

## 4 Experiments

### 4.1 Experimental settings

We use the ACE 2004 corpus to simulate active learning. We treat each of the relation types in turn as the target type to be learned. We collect all pairs of entity mentions appearing in the same sentence to be the candidates for querying. Our task is to find the target relations and obtain reasonable performance using limited hand-labeled data. We use the original tags in the corpus to answer the queries during the active learning process, which simulates hand-labeling. We take

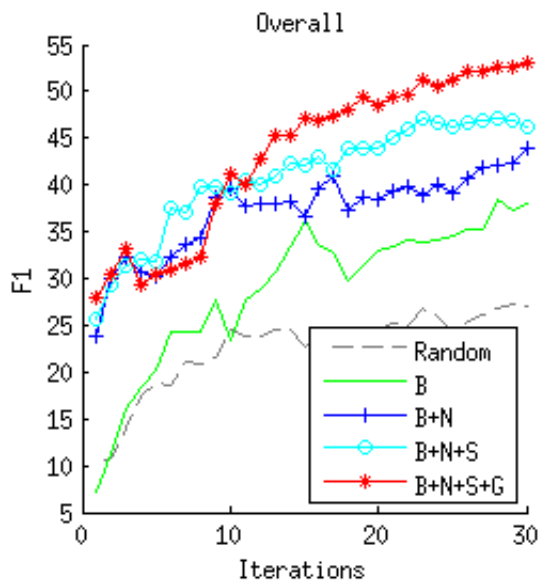


Figure 1. Improvement by different components.

B: Baseline, N: Non relations, S: Interleaving Self-Training, G: Preference for the Global View

randomly selected 4/5 of the corpus as the sampling space for active learning, and the remaining 1/5 as the test set.

### 4.2 Evaluation

We compare our work to the pure co-testing based active learning (Sun and Grishman 2012), and show the F1 measure given the same number of iterations (5 queries per iteration). For random selection of target seeds, we use the same random sequence for both the baseline and our framework for fair comparison. In the co-testing framework, the contention set will be empty at some point, which gives the final model of active learning. We report the overall improvement when the system achieves a reasonable performance with limited human annotation (30 iterations) and the final performance (Table 1). The overall result is the average of the F1 measure of all types.

Even though the initial non-relation selection led to early learning of entity type constraints, during the active learning process, there remain queries that could be answered automatically by entity type and co-reference rule filters. The hand-labeling cost could thereby be further reduced (Table 2). For some sparse types, the reduction by these filters is substantial. In practice, this has to deal with noise from the NE tagger, but is still helpful as long as there is a decent NE tagger.

On the whole, our system substantially outperforms the baseline with a small number of labeled examples (150 instances, at the 30<sup>th</sup> iteration) and also after a relatively large amount has been annotated (the final model)

To show the effectiveness of each component of our framework, we display the overall performance comparison including random sampling, over the first 30 iterations (Figure 1). At this point, most of the six relations have not reached their stopping point, and so the benefits of the individual components are more evident.

The overall F1 score is the direct average of the F1 scores of the six types. Non-relation approximation gives an improvement since auto-labeling a certain number of non-relations saves quite a few queries, and the better initial balance of positive and negative examples also makes the model select more informative queries from the beginning. Self-training boosts the system further as it incorporates more instances (especially positives) automatically. After these, the preference for the global view also gives improvement after 10 iterations. As a trade-off strategy between density and uncertainty, it is common that such methods only outperform the baseline for a certain duration.

With these components and auto-labeling with type constraints (Table 2), we provide a quite reasonable relation extraction system given only 150 labels.<sup>4</sup> With more labels, we can approximate supervised learning. So we can build a relation extraction system quickly when there is no relation annotation in a new corpus. If we need more relations in this new corpus, we can start the framework again, treating previously acquired relations as labeled negative instances of the new target relation. Experiments on this multiple relation type extension also show similar gains over the baseline system using our methods.

---

<sup>4</sup> Keep in mind that the best systems, trained on thousands of examples, only achieve F scores in the low 70's.

## 5 Conclusion

We present a more practically efficient way to do active learning than a pure co-testing based algorithm. The improvement is most pronounced initially, for small numbers of annotations. We can now achieve reasonable performance for extracting relations with very little annotation. Adding a new relation in an hour now seems within reach.

Each component in the framework is still worth further study. We can consider further efforts to enlarge and balance the initial set from the view of non-relation approximation. We can also try more adaptive semi-supervised algorithms to interleave with co-testing. The quality of the global classifier in the co-testing also remains a constraint, so we will be investigating alternative similarity metrics. While the experiments reported here involve simulated active learning, we are now planning real, human-in-the-loop active learning trials.

## Acknowledgment

This research was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058 and via Department of Interior National Business Center (DoI/NBC) contract number D11PC20154. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, DoI/NBC, or the U.S. Government.

## References

- Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Pinar Donmez and Jaime G. Carbonell and Paul N. Bennett. 2007. Dual strategy active learning. *In Proceedings of the European Conference on Machine Learning (ECML)*.
- Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. *In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

- Zornista Kozareva and Eduard Hovy. 2010. Not all seeds are equal: Measuring the quality of text mining seeds. *In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Dan Roth and Kevin Small. 2008. Active learning for pipeline models. *In Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*.
- Min Song, Hwanjo Yu, and Wook-Shin Han. 2011. Combining active learning and semi-supervised learning techniques to extract protein interaction sentences. *BMC Bioinformatics*, **12** (Suppl-12): S4.
- Ang Sun and Ralph Grishman. 2012. Active Learning for Relation Type Extension with Local and Global Data Views. *In Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*.
- Ang Sun, Ralph Grishman and Satoshi Sekine. 2011. Semi-supervised Relation Extraction with Large-scale Word Clustering. *In: Proceedings of HLT '11: the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Hans Uszkoreit. 2011. Learning relation extraction grammars with minimal human intervention: strategy, results, insights and plans. *In Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing (CI-Cling)*.
- Vishnu Vyas, Patrick Pantel, Eric Crestan. 2009. Helping Editors Choose Better Seed Sets for Entity Expansion. *In Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*.
- Hong-Tao Zhang, Min-Lie Huang, Xiao-Yan Zhu. 2012. A Unified Active Learning Framework for Biomedical Relation Extraction. *In J. Comput. Sci. Technol.*, **27** (2012), Nr. 6, S. 1302-1313.
- Yi Zhang. 2010. Multi-Task Active Learning with Output Constraints. *In Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*
- GuoDong Zhou, Jian Su, Jie Zhang, Min Zhang. 2005. Exploring Various Knowledge in Relation Extraction. *In Proceedings of the 43rd Annual meeting of the Association for Computational Linguistics (ACL)*.

# Parsing Dependency Paths to Identify Event-Argument Relations

**Seung-Cheol Baek**

CS Dept., KAIST

291 Daehak-ro, Yuseong-gu, Daejeon,  
305-701, Republic of Korea

scbaek@nlp.kaist.ac.kr

**Jong C. Park**

CS Dept., KAIST

291 Daehak-ro, Yuseong-gu, Daejeon,  
305-701, Republic of Korea

park@nlp.kaist.ac.kr

## Abstract

Mentions of event-argument relations, in particular dependency paths between event-referring words and argument-referring words, can be decomposed into meaningful components arranged in a regular way, such as those indicating the type of relations and the others allowing relations with distant arguments (e.g., coordinate conjunction). We argue that the knowledge about arrangements of such components may provide an opportunity for making event extraction systems more robust to training sets, since unseen patterns would be derived by combining seen components. However, current state-of-the-art machine learning-based approaches to event extraction tasks take the notion of components at a shallow level by using n-grams of paths. In this paper, we propose two methods called pseudo-count and Bayesian methods to semi-automatically learn PCFGs by analyzing paths into components from the BioNLP shared task training corpus. Each lexical item in the learned PCFGs appears in 2.6 distinct paths on average between event-referring words and argument-referring words, suggesting that they contain recurring components. We also propose a grounded way of encoding multiple parse trees for a single dependency path into feature vectors in linear classification models. We show that our approach can improve the performance of identifying event-argument relations in a statistically significant manner.<sup>1</sup>

## 1 Introduction

Event extraction tasks can be viewed as identifying event-argument relations between tokens by mapping events onto tokens, to be called henceforth *triggers*, even though events may have oth-

er events as arguments in contrast to average relation extraction tasks, leading to interdependencies between events. On looking into mentions of event-argument relations, in particular the shortest dependency path between triggers and arguments, one may find that they can be decomposed into intuitively meaningful components arranged in a regular way, such as *core components* indicating the type of relations and *subordinate components* making it possible for events to take arguments further away from triggers (e.g., coordinate conjunction). We anticipate that the knowledge about arrangements of components provides an invaluable opportunity for making event extraction systems more robust to the choice of training sets, for example by assembling seen components into unseen patterns. Towards this goal, we propose in this paper a way of automatically learning and exploiting internal structures of dependency paths for a robust extraction of biological events from the biological literature with the corpora provided by a series of BioNLP shared tasks (Kim et al., 2009 and Kim et al., 2011).

For example, the following sentence has annotated positive regulation events, including the induction of IP-10 by IFN, in the training corpus.

- (1) IL-10 preincubation resulted in the inhibition of gene expression for several IFN-induced genes, such as IP-10, ISG54, and intercellular adhesion molecule-1. (PMID: 10029571)

From this sentence, we may formulate the pattern “X-induced genes, such as Y” with slots X and Y to detect the THEMES of positive regulation events based on the underlined expression. This pattern can also be decomposed into a core component “X-induced Y” and a subordinate component “genes such as Y”. These two components have different roles. That is, the core component

<sup>1</sup> All the datasets and codes used in this study are available at <http://www.biopathway.org/ijcnlp2013>



alone can be used to detect the THEMES of positive regulation events (e.g., “IFN-induced IP-10”), but the subordinate component alone cannot. Core components may not appear together in a pattern, but subordinate components may (e.g., there are two other involved subordinate components “genes such as Y” and “[PROTEIN] and Y”, where “[PROTEIN]” would be replaced with any protein or gene name, in “IFN-induced genes, such as IP-10, ISG54 and intercellular adhesion molecule-1”). From this observation it is possible to come up with an unseen pattern “X-induced Y and Z”.

However, current state-of-the-art machine learning-based approaches exploit the notion of components of patterns only at a shallow level using n-grams encoding partial structures of dependency graphs (including unigrams used in bag-of-words models), not to mention the notion of regularity in arrangements of components (e.g., Björne et al., 2009; Miwa et al., 2010; Riedel et al., 2011). Therefore, their approaches would be biased towards dependency paths that contain a number of components even overlapping with one another, even though such paths may have undesired meanings due to the arrangements of components.

In this paper, we propose two methods (called *pseudo-count* and *Bayesian* methods) to semi-automatically learn three types of probabilistic context-free grammars (PCFGs) that assume different internal structures of paths, with the help of which dependency paths will be analyzed into components. All the learned PCFGs contain lexical items covering an average of about 2.6 distinct paths between triggers and arguments in the training corpus, suggesting that the methods successfully identified recurring components. To exploit multiple parse trees derived from a single path, we also propose a linear classification model whose output score approximates the difference between the log probabilities of the path being derived from positive and negative relations. We find that the use of PCFGs learned by our pseudo-count method improves the performance of classifiers in a statistically significant manner, compared to a baseline classifier with n-grams encoding partial structures of paths.

## 2 Related Work

The literature on information extraction (IE) contains a number of studies in which dependency paths are found to play a significant role (Johansson and Nugues, 2008; Miwa et al., 2010b; Qiu

et al., 2011). Likewise, the biological event extraction research, a branch of information extraction, stresses the importance of the role of dependency paths in identifying event-argument relations due to the resemblance of event-argument relations to dependency relations (Björne et al., 2008). For this reason, most of the event extraction studies have to use dependency path features, such as n-grams ( $n=1\sim 4$ ) of dependencies and words, the length of dependency paths and so on, in identifying event-argument relations (Björne et al., 2009 and Miwa et al., 2010b).

It is thus not surprising that while there are many studies on dependency paths in the IE literature, most of them focus on identifying the type of dependency graph representations that is most suitable to their problem (cf. Johansson and Nugues, 2008, Miwa et al., 2010a), with few exceptions including Kilicoglu and Berger (2009) and Joshi and Penstein-Rosé (2009). In particular, Kilicoglu and Bergler (2009) manually constructed a total of 27 dependency path patterns by examining dependency paths between triggers and arguments. Joshi and Penstein-Rosé (2009) first generated sequences of triples of dependency relations and the baseform/POS of their tokens and then generalized the sequences by concealing one of the elements of the triples. They find that the use of such generalized sequences improves the performance of the task of identifying opinions from product reviews. However, there are no studies on automatically learning and using the internal structure of the dependency paths that express semantic relations between tokens, as addressed in this paper, to the best of our knowledge.

## 3 Problem Setting

Our proposal is tested on the event extraction task as defined in the 2009 BioNLP shared task 1 (Kim et al., 2009), which was later renamed as GENIA Event Task 1 and extended to cover full papers in the 2011 Bio-NLP shared task (Kim et al., 2011). Their task is to extract structured information on events from sentences in the biological literature, including their event type and participants encoded with a controlled vocabulary that has nine event types and two role types (“THEME” and “CAUSE”). This task can be considered to consist of two sub-tasks, one of identifying triggers and another of identifying event-argument relations. In this study, we focus on the latter and use the gold-standard annota-

tions of triggers in the training and development corpora (including full papers) to generate dependency paths for training and testing.

In order to identify event-argument relations, we use twelve binary classifiers for all the possible combinations of event and role types. One may argue that multi-class classifiers are more suitable for this setting than binary classifiers, but there is no conclusive evidence for their advantage (cf. Baek and Park, 2012). Note also that our present focus is on assessing the benefit from the use of the knowledge about internal structures of dependency paths and not on assessing the whole event extraction systems.

## 4 Method

### 4.1 Preparation of Training Sequences

The shortest dependency paths between triggers and argument candidate words (e.g., “-induced” and “IP-10” in (1)) over basic Stanford dependency graphs<sup>2</sup> (de Marneffe et al., 2006) are first computed, from which the three types of sequences are extracted in turn: *token sequences*, or a sequence of the surface forms of the visited tokens (e.g., “induced gene as IP-10”), *dependency sequences*, or a sequence of the visited dependencies (or more precisely, their type and direction; e.g., “-amod +prep +pobj”), and *combined sequences*, or a sequence of the visited tokens and dependencies (e.g., “induced -amod genes +prep as +pobj IP-10”).

Training sequences are derived from the extracted sequences by preprocessing them as follows. First, the last tokens of sequences, namely arguments, are dropped, because of the observation that this makes it easy to convert the components of patterns into sequences and their subsequences in a systematic way. For example, the two components “-induced Y” and “genes, such as Y” of the pattern “-induced genes, such as Y” can be converted into the sequences “-induced -amod” and “genes +prep as +pobj”, which are combined into a sequence corresponding to the pattern, namely, “induced -amod genes +prep as +pobj”. Second, protein names are replaced with a special token “[PROTEIN]” to help learn generalized patterns, since there are a considerable amount of different types of proteins. Third, the first occurrence of each word in the training corpus is replaced with a special token “[UN-

KNOWN]” to simulate encounters with unknown words in the test corpus during learning. Its downside is that all the tokens in the first training sequence are replaced with “[UN-KNOWN]”.

Note that it is a natural extension of our work to additionally generate other types of sequences, for example by replacing the surface forms of tokens with their other attributes (e.g., POSs and surface forms concatenated with POSs) in sequences mentioned above and by dropping functional tokens (e.g., prepositions) within token sequences, even though we do not consider them here.

### 4.2 PCFG Induction

A PCFG consists of production rules (of the form  $A \rightarrow x$ ), each indicating that a nonterminal symbol  $A$  (a *parent symbol*) is replaced with a sequence  $x$  of symbols (*child symbols*) with a predefined probability. Our PCFGs have two types of production rules, those that produce a sequence of nonterminal symbols (*non-lexical production rules*) and the others that produce a lexical item (*lexical production rules*). In our PCFGs, non-lexical production rules are crafted manually and lexical production rules are learned. The probability of each rule is determined by maximum-likelihood estimation (MLE), which divides the total number of the occurrences of the rule in training parse trees by the total number of the occurrences of its parent symbol in training parse trees.

We build two sets of non-lexical rules, one generating positive sequences and another generating negative sequences, together with the following two non-lexical rules, where “S” stands for the start symbol and “Positive” and “Negative” symbols are the ones to be expanded into positive and negative sequences, respectively.

- (2)  $S \rightarrow \text{Positive}$
- (3)  $S \rightarrow \text{Negative}$

We come up with the following three types of non-lexical rules for positive sequences, where the underlined symbols are *lexical symbols*, or the ones to be expanded into single lexical items, and asterisks indicate that the marked symbols may occur zero or more times in a row.

- (4) Unigram Rules  
Positive  $\rightarrow$  Component Component\*
- (5) Uni-directionally Growing Rules  
Positive  $\rightarrow$  Core Component\*

<sup>2</sup> Since arguments and triggers may be hyphenated, we preprocess dependency graphs, so that hyphenated words are separated into their component words.

(6) Bi-directionally Growing Rules  
Positive  $\rightarrow$  Component\* Core Component\*

These rules assume that sequences consist of components that may appear independently of one another (*independence constraint*), but also that they cannot overlap with one another (*non-overlapping constraint*). The second and third types of rules assume that sequences should have core components as indicated by the “Core” symbols. The independence constraint may not capture the nature of dependency paths, but makes it cheaper to learn lexical rules. We leave the question about the effect of the independence constraint open for future research. The uni-directionally growing rules are most consistent with our observation that triggers and their dependencies play a significant role in determining the type of event-argument relations.

Since lexical items are allowed to span across more than one element in positive sequences but are not annotated on training sequences, we need to make a guess at parse trees for each sequence to count the occurrences of rules. To address this problem, we propose two methods. One is called a *pseudo-count method* that assigns all possible parse trees for each training sequence an equal probability (i.e., one divided by the number of all possible parse trees) of the sequence being generated from them, and accumulates the assumed probability (i.e., pseudo-count) of parse trees containing each rule.

Another is called a *Bayesian method* that converts our non-lexical rules into an adaptor grammar, or a description of non-parametric Bayesian models with Chinese Restaurant Processes (CRP) and Pitman-Yor Processes (PYP) (Johnson et al., 2007), by adding production rules, to be called *lexical item production rules*, that replace lexical symbols with a sequence of terminal symbols, such as tokens and dependencies (e.g., “Tokens  $\rightarrow$  Token Token\*”), and by labeling lexical symbols as an *adaptor symbol*, whose expansion to terminal symbols is collected during learning. One advantage of this method is to penalize lengthy lexical items, and as a result, to facilitate analyzing sequences into more than one lexical item, since producing a lengthy lexical item requires the use of many lexical item production rules with a probability below one. In practice, we use the adaptor grammar inference program (Johnson et al., 2007), which samples analyses of input sequences (i.e., sequences of dependency types). We assume that all production rules in our adaptor grammars have the same probability.

We ran two thousand iterations of sampling analyses, but ignored samples during the first half, as these may not be significantly different from randomly assigned initial analyses. Afterwards, we counted the occurrences of lexical items and rules. As a result, 1,000 samples are taken for each sequence.

Since negative training sequences can convey a variety of semantic types, it is unlikely that a training corpus contains all possible negative training sequences covering such semantic types, suggesting the risk of over-fitting of learned PCFGs to negative training sequences (cf. Li et al., 2010). To avoid it, we use a simple grammar, which is expected to be able to learn from a relatively small amount of training instances, as shown below, where “NComponent” symbols produce single token and dependency types. In contrast to the positive sequences, it is straightforward to construct parse trees for negative sequences and to count production rules, since all negative sequences have only one possible parse tree.

(7) Negative  $\rightarrow$  NComponent NComponent\*

Finally, we filter out infrequent and lengthy lexical items, which may have the same form as the sequences from which they are learned, to prevent models from memorizing training sequences as they are (e.g., “induced -amod genes +prep as +pobj”), that is, assigning a high weight to them and to teach instead models ways of analyzing positive sequences into relatively small lexical items (e.g., “induced -amod” and “genes +prep as +pobj”). For each lexical symbol, we remove the least probable lexical items whose occurrences form a predefined percentage<sup>3</sup> of the occurrences of the lexical symbol. Note that it is apparently a more reasonable option to learn PCFGs and linear classification models on two different disjoint subsets of randomly selected sequences. We leave this option for future work.

### 4.3 Linear Classification Model

Using the CKY algorithm with beam search, we generate the most probable  $k$  parse trees for three types of sequences extracted from a dependency path with the help of the learned PCFGs, each of which explicitly has a favorite label. One way to

---

<sup>3</sup> The predefined percentage is 1% if the ratio of the number of distinct sequences to the number of sequences is below a third, 5% if the ratio is between a third and two third, and 10% otherwise.

combine their opinions is to let respective classifiers  $S$  for the types of sequences vote for their favorite label  $z_s(x)$  (+1 or -1) of a path  $x$  and to count their vote with a different weight proportionate to their reported confidence  $w_s(x)$  and their credibility  $c_s$ , as follows:

$$y = \sum_S c_s z_s(x) w_s(x) = \sum_S c_s y_s(x)$$

If the output score  $y$  is positive, our classifier makes a final decision of labeling  $x$  as being positive. The term  $z_s(x)w_s(x)$  can be regarded as the output score  $y_s(x)$  given by a classifier  $S$ .

We define  $y_s(x)$  as follows, where the capital letters stand for random variables:

$$y_s(x) = \log\left(\frac{P(Z = +1, X = x)}{P(Z = -1, X = x)}\right)$$

The log probability  $\log(P(Z, X))$  is written in terms of the probability  $P(Z, T)$  of our PCFGs generating  $T_z$  parse trees supporting a value  $z$  of  $Z$ :

$$\log(P(z, x)) = \log(T_z \times \overline{P(z, T_x)})$$

where  $\overline{P(z, T_x)}$  is the average of the probability of parse trees generating  $x$  and supporting  $z$ . Using Jensen's inequality, it is easy to show that its lower bound  $l(z, x)$  is:

$$\log(P(z, x)) \geq l(z, x) = \overline{\log P(z, T_x)} + \log T_z$$

where the first term is the average of the log probability of parse trees under consideration. One thing to note is that the equality always holds for  $P(Z = -1, X = x)$ , since our PCFGs for negatively labeled sequences produce at most one parse tree for each sequence. For this reason, the lower bound  $y_s^{low}(x)$  of  $y_s(x)$  is:

$$y_s^{low}(x) = \sum_z \left( \sum_{t \rightarrow x|z} \frac{z \log P(T = t)}{T_z} + z \log T_z \right)$$

Instead of  $y_s(x)$ , we use  $y_s^{low}(x)$  at risk of the deterioration of the performance of the resulting model, since it is apparently easier to handle than  $y_s(x)$ .

In the worst case, the difference between  $\log P(Z, X)$  and  $l(z, x)$  can be:

$$|\log(P(z, x)) - l(z, x)| \leq \log \frac{\overline{P(z, T_x)}}{P_{min}(z, T_x)}$$

where the denominator is the least probability of parse trees under consideration. It indicates that with a wide beam width the estimated value of  $y_s^{low}(x)$  may be significantly lower than the true value of  $y_s(x)$ , while with a narrow beam width the estimated value of  $y_s^{low}(x)$  is likely to be similar to the estimated value of  $y_s(x)$ , which may be significantly higher than its true value. Thus the success of the use of  $y_s^{low}(x)$  is dependent on the beam width.

Expanding the log probability  $\log P(T = t)$ ,  $y_s^{low}(x)$  is rewritten:

$$\sum_r \sum_z z \log(p_r) \left( \sum_{t \rightarrow x|z} \frac{\text{count}(r \text{ in } t)}{T_z} \right) + \left( \sum_z z \log T_z \right)$$

where  $p_r$  is the probability of rule  $r$ , which is given by PCFGs, and  $\text{count}(r \text{ in } t)$  is the number of the occurrences of rule  $r$  in parse tree  $t$ . Introducing coefficients  $w_r$ ,  $w_1$  and  $w_0$  into the equation,  $y_s^{low}(x)$  can be generalized to a linear model as shown below.

$$\sum_r w_r \left( \sum_{t \rightarrow x|z} \frac{\text{count}(r \text{ in } t)}{T_z} \right) + w_1 \left( \sum_z z \log T_z \right) + w_0$$

Being their linear combination of the linear models  $y_s^{low}(x)$ , the output score  $y$  is also a linear model. In this paper, we train our linear classifiers using LIBLINEAR (Fan et al., 2008)<sup>4</sup>.

Finally, we note that as in the re-ranking parsers (e.g., Charniak and Johnson, 2005), it is possible to use *global features*, or features not allowed in the CKY algorithm, to calculate the log probability  $\log P(T = t)$ . In this paper, we leave the effect of the use of such global features for future research.

## 5 Experiments

We generated labeled training dependency paths for each event-argument relation type from the BioNLP training corpus with the help of the Charniak-Johnson parser (Charniak and Johnson, 2005) with a self-trained biomedical parsing model (McClosky and Charniak, 2008). There are 7,009 positive paths and 10,603 negative paths. The ratio of the number of positive paths

<sup>4</sup> Our linear classifiers are trained using the L2 regularized logistic regression solver with cost constants that are chosen among 0.01, 0.1, 1 and 100 with the help of five-fold cross validation.

to the number of negative paths is 0.66. We found that a majority of the relation types would have a balanced set of training instances, except for a few relation types with the imbalance between positive and negative instances. One way of correcting the imbalance is to give more weight to positive instances, but we leave out the imbalance in this experiment.

We extracted three types of sequences from them. We found that most distinct negative sequences appear once in the training corpus as shown in Table 1, where the bracketed figures are the ratios of the number of distinct sequences to the number of sequences, justifying the use of a simple grammar for negative sequences.

Sequence	Positive	Negative	Total
Combined	3,703 (1.89)	9,781 (1.08)	13,484 (1.31)
Token	3,366 (2.08)	9,270 (1.14)	12,636 (1.39)
Dependency	1,816 (3.86)	7,419 (1.43)	9,235 (1.91)

Table 1. Distinct Training Sequences

We use the pseudo-count and Bayesian methods to learn grammars. The learned PCFGs contain the mentioned example lexical items, “-induced -amod”, “genes +prep as +pobj” and “[PROTEIN] +conj”. They contain a number of intuitively correct core and subordinate components. The learned subordinate components include “genes +prep like +pobj”, “[PROTEIN] +abbrev” and “[PROTEIN] +appos”.

With three different beam widths, we parse sequences to generate feature vectors for our linear classification models and evaluate the resulting models in terms of accuracy, as shown below.

Grammar	Beam Width		
	k=1	k=10	k=100
Pseudo-Count			
Unigram	<b>86.43%</b>	85.97%	86.07%
Uni-direct	86.94%	<b>87.05%</b>	87.03%
Bi-direct	<b>86.48%</b>	86.43%	86.25%
Bayesian			
Unigram	82.72%	<b>83.39%</b>	83.27%
Uni-direct	82.95%	<b>83.70%</b>	82.88%
Bi-direct	82.70%	<b>83.45%</b>	83.26%

Table 2. Accuracy of Our Classifiers

For each grammar, the best reported accuracy is set in bold. With PCFGs learned by the pseudo-count method, the use of multiple parse trees does not affect or even decrease the performance of classifiers. One possible explanation is that the wider the beam is the more erroneous parse trees are likely to affect the final decision of classifiers. In contrast, the classifiers with PCFGs learned by

the Bayesian model would slightly benefit from the use of multiple parse trees, even though their performance also drops when using the widest beam. To explain that we get only a slight benefit from a wide beam width, we looked at feature vectors, noticing that many positive training sequences have only a small number of possible parse trees. We also observed that as expected, classifiers with the uni-directionally growing PCFGs outperform the other classifiers, with one exception of classifiers with the widest beam and the ones learned by the Bayesian method.

To compare with our classifiers, we implement *linear baseline classifiers* that use as features all n-grams (n=1~4) of token, dependency and combined sequences extracted from the training instances. They first replace unknown words in an input sequence with a special token “[UNKNOWN]” and count the occurrence of n-grams in the sequence. Like our classifiers, they are also trained by LIBLINEAR (Fan et al., 2008).

The accuracy of the baseline classifiers is 85.76%, which is lower than that of the pseudo-count classifiers with any beam width in use, but higher than that of the Bayesian classifiers with any used beam width. The superiority of the pseudo-count classifiers with any beam width over the baseline classifiers is statistically significant at the 10% significance level in terms of their accuracy (p-value=5.6~8.4%), according to the one-sided paired Student’s *t*-test with the accuracy of classifiers for each relation type.

## 6 Conclusion

In this paper we proposed a way of exploiting internal structures of dependency paths for the extraction of biological events from the biological literature with the BioNLP shared task corpora. We proposed pseudo-count and Bayesian methods to learn three types of PCFGs that assume different internal structures of paths from dependency paths. To use multiple parse trees for a single path, we also developed a linear classification model whose output score approximates the difference between the log probabilities of the path being derived from positive and negative relations. Finally, we have shown that our approach can improve the performance of identifying event-argument relation in a statistically significant manner.

## Acknowledgments

This work was supported by the National Research Foundation (NRF) of Korea funded by the Ministry of Education, Science and Technology (MEST) (No. 20110029447). We are also grateful to the anonymous reviewers who helped improve the clarity of the paper. All remaining errors are of course ours.

## References

- Baek, S. C. and Park, J. C. (2012, September) Use of Clue Word Annotations as the Silver-standard in Training Models for Biological Event Extraction. In *Proceedings of the SMBM 2012* (pp. 34-41).
- Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., & Salakoski, T. (2009, June). Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task* (pp. 10-18). Association for Computational Linguistics.
- Björne, J., Pyysalo, S., Ginter, F., & Salakoski, T. (2008, September). How complex are complex protein-protein interactions. In *Proceedings of the SMBM 2008* (pp. 125-128).
- Charniak, E., & Johnson, M. (2005, June). Coarse-to-fine n-best parsing and MaxEnt discriminative re-ranking. In *Proceedings of the ACL 2005* (pp. 173-180). Association for Computational Linguistics.
- de Marneffe, M. C., MacCartney, B., and Manning, C. D. (2006, May). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC* (Vol. 6, pp. 449-454).
- Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9, 1871-1874.
- Johansson, R., and Nugues, P. (2008, August). The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (pp. 393-400). Association for Computational Linguistics.
- Johnson, M., Griffiths, T. L., & Goldwater, S. (2007). Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. *Advances in neural information processing systems*, 19, 641.
- Joshi, M., and Penstein-Rosé, C. 2009. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference* (pp. 313-316). Association for Computational Linguistics.
- Kilicoglu, H. and Bergler, S. 2009. Syntactic Dependency Based Heuristics for Biological Event Extraction. In *Proceedings of the BioNLP Shared Task 2009 Workshop* (pp. 119-127).
- Kim, J. D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. I. (2009, June). Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task* (pp. 1-9).
- Kim, J. D., Wang, Y., Takagi, T., and Yonezawa, A. (2011, June). Overview of GENIA event task in BioNLP shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop* (pp. 7-15).
- Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., & Steedman, M. (2010, October). Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1223-1233). Association for Computational Linguistics.
- Li, X. L., Liu, B., & Ng, S. K. (2010, October). Negative training data can be harmful to text classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 218-228). Association for Computational Linguistics.
- McClosky, D., and Charniak, E. (2008). Self-training for biomedical parsing. In *Proceedings of the ACL 2008* (pp. 101-104). Association for Computational Linguistics.
- Miwa, M., Pyysalo, S., Hara, T., and Tsujii, J. I. (2010a). A comparative study of syntactic parsers for event extraction. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing* (pp. 37-45). Association for Computational Linguistics.
- Miwa, M., Sætren, R., Kim, J. D., & Tsujii, J. I. (2010b). Event extraction with complex event classification using rich features. *Journal of bioinformatics and computational biology*, 8(01), 131-146.
- Qiu, G., Liu, B., Bu, J., and Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1), 9-27.
- Riedel, S., & McCallum, A. (2011, June). Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Proceedings of the BioNLP Shared Task 2011 Workshop* (pp. 46-50). Association for Computational Linguistics.

# Augmentable Paraphrase Extraction Framework

Mei-Hua Chen<sup>ac</sup>, Yi-Chun Chen<sup>b</sup>, Shih-Ting Huang<sup>b</sup>, Jason S. Chang<sup>b</sup>

<sup>a</sup>Institute of Information Systems and Applications and <sup>b</sup>Department of Computer Science, National Tsing Hua University, HsinChu, Taiwan, R.O.C. 30013

<sup>c</sup>Department of Foreign Languages and Literature, Hua Fan University, Taipei, Taiwan, R.O.C. 22301

{chen.meihua, pieyaaa, koromiko1104,jason.jschang}@gmail.com

## Abstract

Paraphrase extraction relying on a single factor such as distribution similarity or translation similarity might lead to the loss of some linguistic properties. In this paper, we propose a paraphrase extraction framework, which accommodates various linguistically motivated factors to optimize the quality of paraphrase extraction. The major contributions of this study lie in the augmentable paraphrasing framework and the three kinds of factors conducive to both semantic and syntactic correctness. A manual evaluation showed that our model achieves more successful results than the state-of-the-art methods.

## 1. Introduction

Paraphrasing provides an alternative way to express an idea using different words. Early work on paraphrase acquisition has been mainly based on either distributional similarity (e.g., Lin and Pantel, 2001) or the pivot-based approach (e.g., Bannard and Callison-Burch, 2005). Both methods have their strengths and limitations. Distributional similarity is capable of extracting syntactically correct paraphrases, but may risk including antonymous phrases as paraphrases. On the other hand, the pivot approach has the advantage of preserving semantic similarity among the generated paraphrases; however, the quality and quantity of the paraphrases closely correlates with the techniques of bilingual phrase alignment.

Considering single factors, existing paraphrasing methods could lose some linguistic properties. In view of this, we attempt to differentiate the importance of the paraphrase

candidates based on various factors. In this paper, we take a graphical view of the paraphrasing issue. To achieve the goal mentioned above, we adopt the Weighted PageRank Algorithm (Xing and Ghorbani, 2004). English phrases are treated as nodes. The edge weights are determined by various factors such as semantic similarity or syntactic similarity between nodes. It means that the performance of the ranked paraphrase candidates depends on the factors we selected and added. In other words, our framework is augmentable and is able to accommodate various factors to optimize the quality of paraphrase extraction.

In this case, we propose three linguistically motivated factors to improve the performance of the paraphrase extraction. Lexical distributional similarity is used to ensure that the contexts in which the generated paraphrases appear are similar whereas syntactic distributional similarity is adopted for the purpose of maintaining the syntactic correctness. Translation similarity, one more factor, is capable of preserving semantic equivalence. These three selected factors adopted together effectively achieve better performance on paraphrase extraction. The evaluation shows that our model achieves more satisfactory results than the state-of-the-art pivot-based methods and graph-based methods.

## 2. Related Work

Several approaches have been proposed to extract paraphrases. Earlier studies have focused on extracting paraphrases from monolingual corpora. Barzilay and Mckeown (2001) determine that the phrases in a monolingual parallel corpus are paraphrases of one another only if they appear in similar contexts. Lin and Pantel (2001) derive

paraphrases using parse tree paths to compute distributional similarity. Another prominent approach to paraphrase extraction is based on bilingual parallel corpora. For example, Bannard and Callison-Burch (2005) propose the pivot approach to extract phrasal paraphrases from an English-German parallel corpus. With the advantage of its parallel and bilingual natures of such a corpus, the output paraphrases preserve semantic equivalence. Callison-Burch (2008) further places syntactic constraints on extracted paraphrases to improve the quality of the paraphrases. Chan et al. (2011) use monolingual distributional similarity to rank paraphrases generated by the syntactically-constrained pivot method.

Recently, some studies take a graphical view of the pivot-based approach. Kok and Brockett (2010) propose the Hitting Time Paraphrase algorithm (HTP) to measure the similarities between phrases. Chen et al. (2012) adopt the PageRank algorithm to find more relevant paraphrases that preserve both meaning and grammaticality for language learners. In this paper, we, similarly, present the state-of-the-art approach as a graph. However, unlike Kok and Brockett (2010), we treat English phrases (instead of multilingual phrases) as nodes. On the other hand, different from Chen et al. (2012), our model is augmentable by involving varied linguistic information or domain knowledge.

### 3. Method

Typically, the state-of-the-art paraphrase extraction models only deal with single factors such as distribution similarity or translation similarity. However, different linguistic factors could facilitate the paraphrase extraction in various ways. With this in mind, we propose an augmentable paraphrase extraction framework based on a graph-based method, which can be modeled with multiple linguistically motivated factors.

In the following section, we describe the graph construction (Section 3.1). Then the paraphrase extraction framework is outlined in Section 3.2. Section 3.3 introduces the three factors we proposed for optimizing the quality of paraphrase extraction. Finally, we utilize the grid search method to fine-tune the parameters of our model.

### 3.1 Graph Construction

We transform the paraphrase generation problem into a graph-based problem. First, we generate a graph  $G \equiv (V, E)$ , in which an English phrase is a node  $v \in V$  and two nodes are connected by an edge  $e \in E$ . A set of paraphrase candidates  $CP = \{cp_1, cp_2, \dots, cp_n\}$  is generated for a query phrase  $q$  from a bilingual corpus based on the pivot method (Bannard and Callison-Burch, 2005). We further generate a set of transitive paraphrases  $CP' = \{cp'_1, cp'_2, \dots, cp'_m\}$  of the phrase  $q$ , namely, paraphrases  $cp_i$  and their paraphrases  $cp'_j$  in the same manner. We truncate the paraphrase candidates whose translation similarities are smaller than the threshold  $\varepsilon'$ ; we also exclude  $cp_i$  that consists only of a stopword or contains  $q$  or is contained in  $q$ . Thus, some noisy paraphrases are easily eliminated.

Consider the example graph for the query phrase “*on the whole*” shown in Figure 1. We first find its set of candidate paraphrases  $CP$ , including “*generally speaking*”, “*in general*”, “*in a nutshell*”, using the pivot-based method mentioned above. Then for each phrase in  $CP$ , we extract the corresponding paraphrases respectively. For example, “*in brief*”, “*broadly speaking*”, “*in general*” are paraphrases of the first phrase “*generally speaking*” in  $CP$ . During the process, we keep the extracted paraphrases whose translation similarities are larger than  $\delta^2$ . By linking the phrases with their transitive paraphrases, the graph  $G$  is created.

### 3.2 Augmentable Paraphrase Extraction Framework

In this sub-section, we propose an augmentable paraphrase extraction framework, which can be modeled by multiple factors. Considering a graph  $G \equiv (V, E)$ , the PageRank algorithm assigns a value  $PR$  to each node as their importance measurement. We further adopt the Weighted PageRank algorithm (Xing and Ghorbani, 2004) to state the relatedness between nodes. We calculate the weight  $W$  of the edge which links node  $v$  to node  $u$  using various factor functions  $\mathcal{F}_i$ , the weight function is described as follow,

$$W(u, v) = \sum_{i=1}^N \lambda_i \mathcal{F}_i(u, v, q)$$

<sup>1</sup> We set  $\varepsilon=0.01$ .

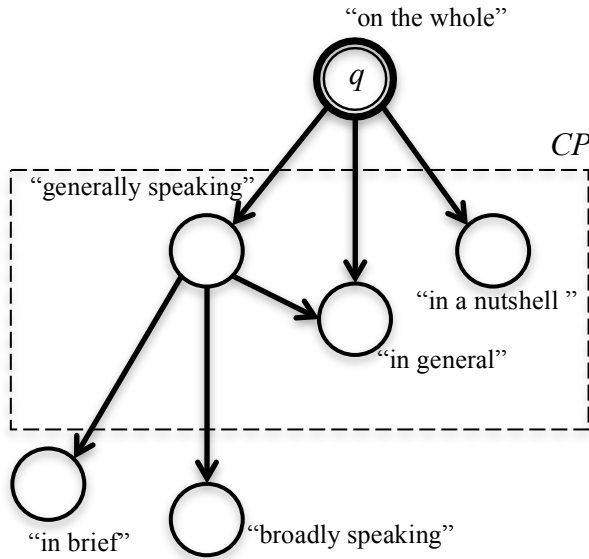
<sup>2</sup> We set  $\delta = 0.0001$ .



where  $q$  is a query phrase,  $\mathcal{F}_i(u, v, q)$  is a factor function and  $\lambda_i$  is the weight of the factor. The weighted  $PR$  value of a certain node  $u$  is defined iteratively as:

$$PR(u) = \sum_{v \in R(v)} PR(v)W(u, v)$$

where  $R(v)$  is a set of nodes that point to  $u$ .



**Figure 1.** Example graph for the phrase “on the whole”.

### 3.3 Linguistically Motivated Factors

Our model enables linguistically motivated factors to optimize the performance of paraphrase extraction. In this sub-section, we introduce three decisive factors: lexical distributional similarity, syntactic distributional similarity and translation similarity.

#### Lexical distributional similarity factor

Lexical distributional information is to ensure that the contexts in which the generated paraphrases appear are similar. For each phrase  $p$  in  $G$ , we extract three kinds of context vectors,  $v_L$ ,  $v_R$ ,  $v_{LR}$  and calculate vector similarities. Vectors  $v_L$  and  $v_R$  represent two sets of adjacent words which occur in the left and right of  $p$  respectively. Words appear simultaneously in both left and right sides of  $p$  are also extracted as the feature vector  $v_{LR}$ . Each item in vectors is an associated score calculated by pointwise

mutual information of the phrase  $p$  (Cover and Thomas, 1991).

Given the query phrase  $q$ , for each paraphrase candidate  $u$  in  $G$ , we calculate the cosine similarity of the context vectors,  $v_L$ ,  $v_R$ ,  $v_{LR}$  between  $q$  and  $u$ . That is, three factors  $\mathcal{F}_{v_L}$ ,  $\mathcal{F}_{v_R}$  and  $\mathcal{F}_{v_{LR}}$  are described as a cosine similarity function:

$$\mathcal{F}_k = \frac{v_{u_k} \cdot v_{q_k}}{|v_{u_k}| |v_{q_k}|}$$

where  $v_{u_k}$  denotes a context vector of  $u$ , and  $v_{q_k}$  a context vector of  $q$  and  $k \in \{L, R, LR\}$ .

#### Syntactic distributional similarity factor

Calculating the extrinsic syntactic similarity between nodes is used to maintain the syntactic correctness of the generated paraphrases. For each phrase  $p$ , we extract three vectors  $s_L$ ,  $s_R$ ,  $s_{LR}$ , which represents the <POS tag, frequency> pairs that appear on the left, right and both left and right sides of the phrase  $p$ . We use the GENIA tagger to obtain POS tags surrounding the phrase  $p$ . Each item in vectors is paired with the frequency of the corresponding tag. For each paraphrase candidate  $u$  of the query phrase  $q$ , we calculate the similarities  $\mathcal{F}_{s_L}$ ,  $\mathcal{F}_{s_R}$  and  $\mathcal{F}_{s_{LR}}$  between the vectors of  $u$  and  $q$  using cosine similarity.

$$\mathcal{F}_k = \frac{s_{u_k} \cdot s_{q_k}}{|s_{u_k}| |s_{q_k}|}$$

where  $s_{u_k}$  denotes a vector of  $u$ , and  $s_{q_k}$  a vector of  $q$ , and  $k \in \{L, R, LR\}$ .

#### Translation similarity factor

Next, we calculate the intrinsic translation similarity which is capable of preserving semantic equivalence. Translation similarity factor for an edge connecting node  $v$  and  $u$  is defined as:

$$\mathcal{F}_{tran} = \sum_{f \in T(v)} P(f|v)P(u|f)$$

where  $u$  is one paraphrase of phrase  $v$ ,  $T(v)$  denotes a set of the foreign-language alignment of  $v$ , and  $P(\cdot)$  the translation probability. Both of the alignment and translation probability are described in Och and Ney (2003).

### 3.4 Parameter Optimization

Once the factors are selected, we have to determine the weights of the factors, (i.e.,  $\lambda_i$  in Section 3.2). In other words, we train the weights of factors such that the performance is optimal for a given developing data set. We use Discounted Cumulative Gain (DCG) (Järvelin and Kekäläinen, 2002) to measure the quality of paraphrases. From the top to the bottom of the result list, the DCG score is accumulated with the gain of each result discounted at lower ranks. The DCG score is defined as:

$$DCG(r, c) = \sum_{i=1}^k \frac{2^{score_i} - 1}{\log_2(i + 1)}$$

where  $r$  represents a set of manually labeled paraphrase scores,  $c$  is a set of paraphrases to be evaluated, and  $score_i$  is the paraphrase score at rank  $i$  of  $c$ .

The parameters<sup>3</sup> are selected in order to maximize the DCG scores in a total of  $S$  query phrases from the developing data set:

$$\hat{\lambda}_1^N = arg \max_{\lambda_1^N} \left\{ \sum_{s=1}^S DCG(r_s, \hat{c}(p_s, \lambda_1^N)) \right\}$$

where  $\hat{c}$  is a set of paraphrases of the query phrase  $p_s$ , extracted from our model under the parameter values  $\lambda_1^N$ .

In the process, we first assign each parameter a random value ranging from 0 to 1 and use a grid-based line optimization method to optimize the parameters. While optimizing a parameter, we maximize the parameter of certain dimension while the parameters of other dimensions are fixed. The process stops when the values of the parameters do not change in two iterations.

## 4. Results

### 4.1 Experimental Setting

In this paper, we adopted the Danish-English section (containing 1,236,427 sentences) of the Europarl corpus, version 2 (Koehn, 2002) for computing distributional similarity and translation similarity. Word alignments were

<sup>3</sup> In this paper, the parameters are  $\lambda_{v_L} = 0.03$ ,  $\lambda_{v_R} = 0.01$ ,  $\lambda_{v_{LR}} = 0.99$ ,  $\lambda_{s_L} = 0.00001$ ,  $\lambda_{s_R} = 0.00001$ ,  $\lambda_{s_{LR}} = 0.18$  and  $\lambda_{tran} = 0.06$ .

produced by Giza++ toolkit (Och and Ney, 2003). We randomly selected 50 phrases as the developing set for optimizing parameters. For each phrase, three distinct sentences which containing the phrase are randomly sampled. A total of 6073 paraphrases have been labeled score 0 (incorrect), 1 (partially correct), and 2 (correct) by considering the fluency of each sentence for developing optimization.

We compared our augmentable paraphrase extraction framework (**APF**) with three baselines: the syntactically-constrained pivot method (**SBiP**) (Callison-Burch, 2008), syntactically-constrained pivot method using monolingual distributional similarity (**SBiP-MonoDS**) (Chan et al., 2011) and the graph-based method (**GB**) (Chen et al., 2012). To assess the contribution of the parameter optimization, we built another model based on APF with identical weights of factors (**APF-avgW**).

We evaluated the paraphrase quality through a substitution test. We randomly selected 133 most commonly used phrases from 30 research articles. For each phrase, we extracted the corresponding paraphrase candidates and evaluated its top 5 candidates. At the same time, three or less distinct sentences containing the phrase were randomly sampled (a total of 398 sentences were evaluated) from the New York Times section of the English Gigaword (LDC2003T05) to capture the fact that paraphrases are valid in some contexts but not others (Szpektor et al., 2007). Two native speaker judges evaluated whether the candidates are syntactically and semantically appropriate in various contexts. They assigned two values corresponding to the semantic and syntactic considerations to each sentence by score 0, (not acceptable), 1 (“acceptable”) and 2 (“acceptable and correct”). The inter-annotator agreement was 0.67.

It is worth noting that we include two measurement schemes for comprehensive analysis. The strict scheme considers a paraphrase as “correct” if and only if both of the two judges scored 2 points, whereas the other one considers a paraphrase as “acceptable” if it is given scores of 1 or 2.

### 4.2 Experimental Results

We compared the performance of the five models, **SBiP**, **SBiP-MonoDS**, **GB**, **APF-avgW** and **APF**, using the precision, coverage, MRR and DCG. Because the number of paraphrases generated by **SBiP**, **SBiP-DS** (101 phrases) and **GB**, **APF-avgW**, **APF** (131 phrases) are varied, we

decided to analyze the results of 99 phrases involving 295 sentences which were generated by all five models. Top- $k$  precision indicates the percentage of the sentences in which correct paraphrase(s) appear in the top- $k$  paraphrase candidates. The coverage was measured by the number of sentences in which at least one out of five paraphrases is correct within all 398 sentences.

Table 1 shows the results of precision and coverage in overall consideration. As can be seen, the **APF** achieved higher precision and coverage than the other four methods.

Additionally, we evaluated the results using MRR. MRR is defined as a measure of how much effort needed for a user to locate the first appropriate paraphrase for the given phrase in the ranked list of paraphrases. As shown in Table 2, the **APF** model performed better than the other models in both correct and acceptable measures. Moreover, Table 3 showed that the **APF** model outperformed the other models in both correct and acceptable measures based on either overall or individual consideration. DCG comprehensively considers both the number of good quality paraphrases and the ranking of these paraphrases. Overall, the **APF** model achieved better performance in paraphrase extraction.

	Top-1 precision	Top-5 precision	Coverage
SBiP	0.13/0.29	0.29/0.59	0.22/0.45
SBip-DS	0.15/0.36	0.32/0.57	0.25/0.44
GB	0.15/0.33	0.29/0.54	0.26/0.51
APF-avgW	0.14/0.39	0.36/ <b>0.66</b>	0.34/ <b>0.61</b>
APF	<b>0.16/0.42</b>	<b>0.38/0.65</b>	<b>0.36/0.61</b>

**Table 1.** Performance of the five models. Note that the former value indicates **correct** measures and the latter one **acceptable** measures.

	Semantic	Syntactic	Both
SBiP	0.19/0.41	0.45/0.52	0.18/0.40
SBip-DS	0.22/0.46	0.48/0.54	0.22/0.45
GB	0.21/0.43	0.51/0.54	0.21/0.42
APF-avgW	0.22/0.50	0.57/0.64	0.21/0.49
APF	<b>0.24/0.51</b>	<b>0.58/0.65</b>	<b>0.23/0.50</b>

**Table 2.** MRR scores of the five models. Note that the former value indicates **correct** measures and the latter **acceptable** measures.

	Semantic	Syntactic	Both
SBiP	0.24/0.68	0.75/0.89	0.23/0.64
SBip-DS	0.29/0.78	0.86/1.01	0.29/0.73
GB	0.27/0.81	0.96/1.09	0.26/0.75
APF-avgW	0.31/0.93	1.12/1.28	0.31/0.90
APF	<b>0.33/0.96</b>	<b>1.14/1.31</b>	<b>0.33/0.93</b>

**Table 3.** DCG scores of the five models. Note that the former value indicates **correct** measures and the **latter** acceptable measures.

## 5. Conclusion

In this paper, we propose a paraphrase extraction framework. Accommodating various linguistically motivated factors, the framework is capable of extracting better paraphrases carrying linguistic features. The results of the manual evaluation demonstrated that the proposed methods achieved performance improvement in terms of precision, coverage, MRR and DCG. The optimized parameters show that the lexical and syntactic distributional similarity factors make a substantial contribution to our model. Specifically, the words as well as the POS tags appear in both left and right sides show satisfactory performance.

However, some further analyses could be conducted in the future. Although the weights of parameters carry the linguistic properties, the proposed factors could be considered separately for examining and comparing the individual effectiveness in our framework. On the other hand, other factors could be taken in consideration. For example, parsing information could be added to the framework to investigate whether or to what extent it contributes to the paraphrasing task.

## References

- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*, pp. 597-604.
- Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the ACL*, pp. 50-57.
- Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*, pp. 196-205.

- Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pp. 33-42.
- Mei-Hua Chen, Shih-Ting Huang, Chung-Chi Huang, Hsien-Chin Liou and Jason S. Chang. 2012. PREFER: Using a Graph-Based Approach to Generate Paraphrases for Language Learning. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pp. 80-85.
- Thomas M. Cover, Joy A. Thomas. 1991. *Elements of Information Theory*. John Wiley & Sons.
- Jane Frodesen. 2002. Developing paraphrasing skills: A pre-paraphrasing mini-lesson. Retrieved September 19, 2012 from [www.ucop.edu/dws/lounge/dws\\_ml\\_pre\\_paraphrasing.pdf](http://www.ucop.edu/dws/lounge/dws_ml_pre_paraphrasing.pdf).
- Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*. 20(4), pp. 422-446.
- Philipp Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. Unpublished, <http://www.isi.edu/~koehn/europarl/>.
- Stanley Kok and Chris Brockett. 2010. Hitting the right paraphrases in good time. In *Proceedings of NAACL/HLT*, pp. 145-153.
- Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 323-328.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1): 19-51.
- Idan Szpektor, Eyal Shnarch, and Ido Dagan. 2007. Instance based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp.456-463.
- Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Proceedings of the 2nd Annual Conference on Communication Networks and Services Research*, pp. 305-314.

# Automatic Prediction of Evidence-based Recommendations via Sentence-level Polarity Classification

Abeed Sarker , Diego Mollá-Aliod

Centre for Language Technology  
Macquarie University  
Sydney, NSW 2109

{abeed.sarker, diego.molla-aliiod}@mq.edu.au

Cécile Paris

CSIRO – ICT Centre  
Sydney, NSW 2122

cecile.paris@csiro.au

## Abstract

We propose a supervised classification approach for automatically determining the polarities of medical sentences. Our polarity classification approach is context-sensitive, meaning that the same sentence may have differing polarities depending on the context. Using a set of carefully selected features, we achieve 84.7% accuracy, which is significantly better than current state-of-the-art for the polarity classification task. Our analyses and experiments on a specialised corpus indicate that automatic polarity classification of *key* sentences can be utilised to generate evidence-based recommendations.

## 1 Introduction

Evidence Based Medicine is a practice that requires practitioners to rely on the best available medical evidence when answering clinical queries. While this practice improves patient care in the long run, it poses a massive problem of information overload to practitioners because of the large volume of medical text available electronically (e.g., MEDLINE<sup>1</sup> indexes over 22 million articles). Research has shown that the act of searching for, appraising, and synthesising evidence from multiple documents generally requires more time than practitioners can devote (Ely et al., 1999). As a result, practitioners would benefit from automatic systems that help perform these tasks and generate *bottom-line recommendations*.

In this paper, we take the first steps towards the generation of bottom-line, evidence-based summaries. Our analyses reveal that the polarities of *key* sentences in medical documents can be utilised to determine final recommendations associated with a query. *Key* sentences refer to the

<sup>1</sup><http://www.ncbi.nlm.nih.gov/pubmed>

most important sentences in a medical abstract that are associated with a posed query. In our work, we use the sentences extracted by a domain-specific, query-focused text summariser. Consider the following sentence for example:

*A significant body of evidence supports the use of long-acting bronchodilators and inhaled corticosteroids in reducing exacerbations in patients with moderate to severe COPD.*

The sentence is taken from a medical abstract, and clearly recommends the use of *bronchodilators and inhaled corticosteroids*, which are the context interventions in this case. In other words, it has a positive polarity for this task. Since positively polarised *key* sentences generally represent the recommendations, we attempt to automatically identify the polarities of medical sentences as the first step towards generating bottom-line recommendations. We show that sentence-level polarity classification is a useful approach for generating evidence-based recommendations. We model the problem of sentence polarity classification as a binary classification problem, and we present a supervised machine learning approach to automatically classify the polarities of *key* sentences. Our classification approach is context dependent, i.e., the same sentence can have differing polarities depending on the context.

## 2 Related Work

Research work most closely related to ours is that by Niu *et al.* (2005; 2006). In their approach, the authors attempt to perform automatic polarity classification of medical sentences into four categories, and apply supervised machine learning to solve the classification problem. In contrast, our approach takes into account the possibility of the same sentence having multiple polarities. This can happen when multiple interventions are mentioned

in the same sentence, with differing results associated with each intervention. Keeping the end-use of this task in mind, we model the problem as a binary classification problem. We use the approach proposed by Niu *et al.* (2005) as a benchmark approach for comparison, and also use some of the features proposed by them.

The majority of the work related to polarity classification has been carried out outside the medical domain, under various umbrella terms such as: sentiment analysis (Pang *et al.*, 2002; Pang and Lee, 2004), semantic orientation (Turney, 2002), opinion mining (Pang and Lee, 2008), subjectivity (Lyons, 1981) and many more. All these terms refer to the general method of extracting polarity from text (Taboada *et al.*, 2010). The pioneering work in sentiment analysis by Pang *et al.* (2002) utilised machine learning models to predict sentiments in text, and their approach showed that SVM classifiers (Vapnik, 1995) trained using bag-of-words features produced good accuracies. Following this work, such classification approaches have been applied to texts of various granularities: documents, sentences, and phrases. Research has also focused on classifying polarities relative to contexts (Wilson *et al.*, 2009). However, only limited research has taken place on applying polarity classification techniques on complex domains such as the medical domain (Niu *et al.*, 2005; Sarker *et al.*, 2011).

Our aim is to investigate the possibility of using sentence-level polarity classification to generate bottom-line, evidence-based summaries. While there has been some research on automatic summarisation in this domain (Lin and Demner-Fushman, 2007; Niu *et al.*, 2006; Sarker *et al.*, 2013; Cao *et al.*, 2011), to the best of our knowledge, there is no system that currently produces bottom-line, evidence-based summaries that practitioners can utilise at point of care.

### 3 Data, Annotation and Analysis

We use the corpus by Mollá and Santiago-Martinez (2011), which consists of 456 clinical questions, sourced from the Journal of Family Practice<sup>2</sup> (JFP). Each question is associated with one or more bottom-line answers (multi-document summaries) authored by contributors to JFP. Each bottom-line answer is in turn associated with detailed explanations provided by the JFP contrib-

<sup>2</sup><http://www.jfponline.com>

utors; these detailed explanations are generally single-document summaries. The corpus also contains abstracts of source documents that provide the evidence of the detailed explanations.

The bottom-line summaries in the corpus present final recommendations in response to the queries. For example, a bottom-line summary may or may not recommend an intervention in response to a disorder. Thus, the bottom-line summaries can be considered to be polarised — when an intervention is recommended, the polarity is positive, and when it is not recommended, the polarity is non-positive. The bottom-line summaries are generated by synthesising information from individual documents. Therefore, it is likely that the polarities of the individual documents, or their summaries, agree with the polarities of the associated bottom-line summaries.

For the preliminary annotation and analysis, we used the same data as the task-oriented coverage analysis work described in (Sarker *et al.*, 2012). The data consists of 33 manually identified questions. All these questions are treatment questions and the bottom-line summaries mention one or more interventions, some of which are recommended while the others are not. We first annotated the polarities of the bottom-line answers relative to the interventions mentioned. We used two categories for the annotation — recommended/not recommended (positive/non-positive). Figure 1 presents a question, the associated bottom-line summary, and our contextual polarity annotation. All the answers to the 33 questions were annotated by the first two authors of this paper. In almost all the cases, there was no disagreement between the annotators; the few disagreements were resolved via discussion.

Next, we collected the *key* (summary) sentences from the abstracts associated with the bottom-line summaries. To collect the *key* sentences from the documents, we used the QSpec summariser (Sarker *et al.*, 2013), which has been shown to generate content-rich, extractive, three-sentence summaries. We performed polarity annotation of these summary sentences. Similar to our bottom-line summary annotation process, for a sentence, we first identified the intervention(s) mentioned, and then categorised their polarities. We came across sentences where two different interventions were mentioned and the polarities associated with them were opposite. Consider the following sentence

**Question:** *What is the most effective beta-blocker for heart failure?*

**Bottom-line answer:** *Three beta-blockers-carvedilol, metoprolol, and bisoprolol-reduce mortality in chronic heart failure caused by left ventricular systolic dysfunction, when used in addition to diuretics and angiotensin converting enzyme (ACE) inhibitors.*

**Contextual Polarities:** *carvedilol – recommended; metoprolol – recommended; bisoprolol – recommended.*

Figure 1: Sample bottom-line summary and an example of polarity annotation.

fragment, for example:

*The present study demonstrated that the combination of cimetidine with levamisole is more effective than cimetidine alone and is a highly effective therapy ...*

For this sentence, the combination therapy is recommended over monotherapy with *cimetidine*. Therefore, the polarities are: *cimetidine with levamisole* – recommended; *cimetidine alone* – not recommended. At the same time, in a number of cases, although a sentence is polarised, it does not mention an intervention. Such sentences were annotated of this paper without adding any intervention to the context. In this manner, we annotated a total of 589 sentences from the QSpec summaries associated with the 33 questions. If a sentence contained more than one intervention, we added an annotated instance for each intervention.

A subset of the QSpec sentences, 124 in total, were annotated by the second author of this paper and these annotations were used to measure agreement among the annotators. We used the Cohen’s Kappa (Carletta, 1996) measure to compute inter-annotator agreement. We obtained an agreement of  $\kappa = 0.85$ , which can be regarded as almost perfect agreement (Landis and Koch, 1977).

Following the annotation process, we compared the annotations of the single document summary sentences with the bottom-line summary annotations. Given that a summary sentence has been annotated to be of positive polarity with an intervention in context, we first checked if the drug name (or a generalisation of it) is also mentioned in the bottom-line summary. If yes, we checked the polarity of the bottom-line summary. In this

manner, we collected a total of 177 summary sentence – bottom-line summary pairs. Among these, in 169 (95.5%) cases, the annotations were of the same polarity. In the rest of the 8 cases, the QSpec summary sentence recommended a drug, but the bottom-line summary did not.

We also manually examined the 8 cases where there were disagreements. In all the cases, this was either because individual documents presented contrasting results, i.e., the positive findings of one study were negated by evidence from other studies; or because a summary sentence presented some positive outcomes, but side effects and other issues were mentioned by other summary sentences, leading to an overall negative polarity.

If automatic sentence-level polarity classification techniques are to be used for generating bottom-line summaries in a two-step summarisation process, the first step (QSpec summaries) also needs to have very good recall. The QSpec summary sentences contained 99 out of the 109 unique interventions, giving a recall of 90.8%. We examined the causes for unrecalled interventions and found that of the 10 not recalled, 4 were due to missing abstracts from the corpus, and 2 drug names were not mentioned in any of the referenced abstracts. Thus, the actual recall is 96.1%. Considering the high recall of interventions in the summary sentences, and the high agreement among the summary sentences and bottom-line summary sentences, it appears that automatic polarity classification techniques have the potential to be applied for the task of bottom-line summary generation in a two-step summarisation process.

## 4 Automatic Polarity Classification

We model the problem of sentence level polarity classification as a supervised classification problem. We utilise the annotated contexts in our supervised polarity classification approach by deriving features associated with those contexts. We annotated a total of 2362 *key* sentences (QSpec summaries) from the corpus (1736 non-positive and 626 positive instances). We build on the features proposed by existing research on sentence level polarity classification and introduce some context-specific and context-independent features. The following is a description of the features.

### (i) Word n-grams

Our first feature set is word n-grams ( $n = 1$  and  $2$ ) from the sentences. Cues about the polarities

of sentences are primarily provided by the lexical information in the sentences (e.g., words and phrases). We lowercase the words, remove stopwords and stem the words using the Porter stemmer (Porter, 1980). For each sentence that has an annotated context, we replace the context word(s) using the keyword ‘\_CONTEXT\_’. Furthermore, we replace the disorder terms in the sentences using the keyword ‘\_DISORDER\_’. We used the MetaMap<sup>3</sup> tool (Aronson, 2001) to identify broad categories of medical concepts, known as the UMLS<sup>4</sup> *semantic types*, and chose terms belonging to specific categories as the disorders<sup>5</sup>.

#### (ii) Change Phrases

We use the Change Phrases features proposed by Niu *et al.* (2005). The intuition behind this feature set is that the polarity of an outcome is often determined by how a change happens: if a *bad* thing (e.g., mortality) was *reduced*, then it is a positive outcome; if a *bad* thing was *increased*, then the outcome is negative. This feature set attempts to capture cases when a *good/bad* thing is *increased/decreased*. We first collected the four groups of *good*, *bad*, *more*, and *less* words used by Niu *et al.* (2005). We augmented the list by adding some extra words to the list which we expected to be useful. In total, we added 37 *good*, 17 *bad*, 20 *more*, and 23 *less* words. This feature set has four features: MORE-GOOD, MORE-BAD, LESS-GOOD, and LESS-BAD. The following sentence exhibits the LESS-BAD feature, indicating a positive polarity.

*Statistically and clinically significant improvement, including a statistically significant reduction in mortality, has been noted in patients receiving ...*

To extract the first feature, we applied the approach by Niu *et al.* (2005): a window of four words on each side of a MORE-word in a sentence was observed. If a GOOD-word occurs in this window, then the feature MORE-GOOD is activated. The other three features were activated in a similar way. The features are represented using a binary vector with 1 indicating the presence of a feature and 0 indicating absence.

<sup>3</sup><http://metamap.nlm.nih.gov/>

<sup>4</sup><http://www.nlm.nih.gov/research/umls/>

<sup>5</sup>Semantic types in this category: pathological function, disease or syndrome, mental or behavioral dysfunction, cell or molecular dysfunction, virus, neoplastic process, anatomic abnormality, acquired abnormality, congenital abnormality and injury or poisoning

#### (iii) UMLS Semantic Types

We used all the UMLS *semantic types* (identified using MetaMap) present in a sentence as features. Intuitively, the occurrences of *semantic types*, such as *disease or syndrome* and *neoplastic process*, may be different in different polarity of outcomes. Overall, the UMLS provides 133 *semantic types*, and we represent this feature set using a binary vector of size 133 – with 1 indicating the presence and 0 indicating the absence of a *semantic type*.

#### (iv) Negations

Negations play a vital role in determining the polarity of the outcomes presented in medical sentences. To detect negations, we apply three different techniques. In our first variant, we detect the negations using the same approach as (Niu *et al.*, 2005). In their simplistic approach, the authors use the *no* keyword as a negation word and use that for detecting negated concepts. To extract the features, all the sentences in the data set are first parsed by the Apple Pie parser<sup>6</sup> to get phrase information. Then, in a sentence containing the word *no*, the noun phrase containing *no* is extracted. Every word in this noun phrase except *no* itself is attached a ‘NO’ tag. We use a similar approach, but instead of the Apple Pie parser, we use the GENIA Dependency Parser (GDep)<sup>7</sup> (Sagae and Tsujii, 2007), since it has been shown to give better performance with medical text.

For the second variant, we use the negation terms mentioned in the BioScope corpus<sup>8</sup> (Vincze *et al.*, 2008), and apply the same strategy as before, using the GDep parser again. For the third variant, we use the same approach using the negation terms from NegEx (Chapman *et al.*, 2001).

#### (v) PIBOSO Category of Sentences

Our analysis of the QSpec summary sentences suggested that the class of a sentence may be related to the presence of polarity in the sentence. For example, a sentence classified as *Outcome* is more likely to contain a polarised statement than a sentence classified as *Background*. Therefore, we use the PIBOSO classifications of the sentences as a feature. The sentences are classified using the system proposed by Kim *et al.* (2011) into the categories: Population, Intervention, Background, Outcome, Study Design and Other.

<sup>6</sup><http://nlp.cs.nyu.edu/app/>

<sup>7</sup><http://people.ict.usc.edu/~sagae/parser/gdep/>

<sup>8</sup><http://www.inf.u-szeged.hu/rgai/bioscope>



#### (vi) Synset Expansion

Certain terms play an important role in determining the polarity of a sentence, irrespective of context (e.g., some of the *good* and *bad* words used in the *change phrases* feature). Certain adjectives, and sometimes nouns and verbs, or their synonyms, are almost invariably associated with positive or non-positive polarities. Thus, for each adjective, noun or verb in a sentence, we use WordNet<sup>9</sup> to identify the synonyms of that term and add the synonymous terms, attached with the ‘SYN’ tag, as features.

#### (vii) Context Windows

This is the first of our context sensitive features. We noticed that, in a sentence, the words in the vicinity of the context-intervention may provide useful information regarding the polarity of the sentence relative to that drug. Thus, we collect the terms lying inside 3-word boundaries before and after the context-drug term(s). This feature is useful when there are direct comparisons between two interventions. We tag the words appearing before an intervention with the ‘BEFORE’ tag and those appearing after with the ‘AFTER’ tag, and use these as features.

#### (viii) Dependency Chains

In some cases, the terms that influence the polarity of a sentence associated with an intervention do not lie close to the intervention itself, but is connected to it via dependency relationships, and to capture them, we use the parses produced by the GDep parser. For each intervention appearing in a sentence, we identify all the terms that are connected to it via specific dependency chains using the following rule:

1. Start from the intervention and move up the dependency tree till the first VERB item the intervention is dependent on, or the ROOT.
2. Find all items dependent on the VERB item (if present) or the ROOT element.

All the terms connected to the context term(s) via this relationship are collected, tagged using the ‘DEP’ keyword and used as features.

#### (ix) Other Features

We use a number of simple binary and numeric features, which are: context-intervention position, summary sentence position, presence of modals, comparatives, and superlatives.

<sup>9</sup><http://wordnet.princeton.edu/>

## 4.1 Classification, Results and Discussion

In our experiments, we use approximately 85% of our annotated data (2008 sentences) for training and the rest (354 sentences) for evaluation. We performed preliminary 10-fold cross validation experiments on the training set using a range of classifiers and found SVMs to give the best results, in agreement with existing research in this area. We use the SVM implementation provided by the Weka machine learning tool<sup>10</sup>.

Table 1 presents the results of our polarity classification approach. The overall accuracy obtained using various feature set combinations is shown, along with the 95% confidence intervals<sup>11</sup>, and the f-scores for the positive and non-positive classes. The first set of features shown on the table represent the features used by Niu *et al.* (2006); we consider the scores achieved by this system as the baseline scores. The second row presents the results obtained using all context-free features. It can be seen from the table that the two context-free feature sets, expanded synsets and PIBOSO categories, improve classification accuracy from 76% to 78.5%. This shows the importance of these context-free features. All three negation detection variants give statistically significant increases in accuracy compared to the baseline.

The non-positive class f-scores are much higher than the positive class f-scores. The highest f-score obtained for the positive class is 0.74, and that for the non-positive class is 0.89. This is perhaps due to the fact that the number of training examples for the latter class is more than twice to that of the positive class. We explored the effect of the size of training data on classification accuracy by performing more classification experiments. We used different sized subsets of the training set: starting from 5% of its original size, and increasing the size by 5% each time. To choose the training data for each experiment, we performed random sampling with no replacement. Figure 2 illustrates the effect of the size of the training data on classification accuracies.

As expected, classification accuracies and f-scores increase as the number of training instances increases. The increase in the f-scores for the positive class is much higher than the increase for the non-positive class f-scores. This verifies that the

<sup>10</sup><http://www.cs.waikato.ac.nz/ml/weka/>

<sup>11</sup>Computed using the `binom.test` function of the R statistical package (<http://www.r-project.org/>)

Feature sets	Accuracy (%)	95% CI	Positive f-score	Non-positive f-score
i,ii,iii, and iv (Niu et al., 2006)	76.0	71.2 – 80.4	0.58	0.83
Context-free (i-vi)	78.5	73.8 – 82.8	0.64	0.85
All (Niu)	83.9	79.7 – 87.6	0.71	0.89
All (Bioscope)	84.7	80.5 – 88.9	0.74	0.89
All (NegEx)	84.5	80.2 – 88.1	0.73	0.89

Table 1: Polarity classification accuracy scores, 95% confidence intervals, and class-specific f-scores for various combinations of feature sets.

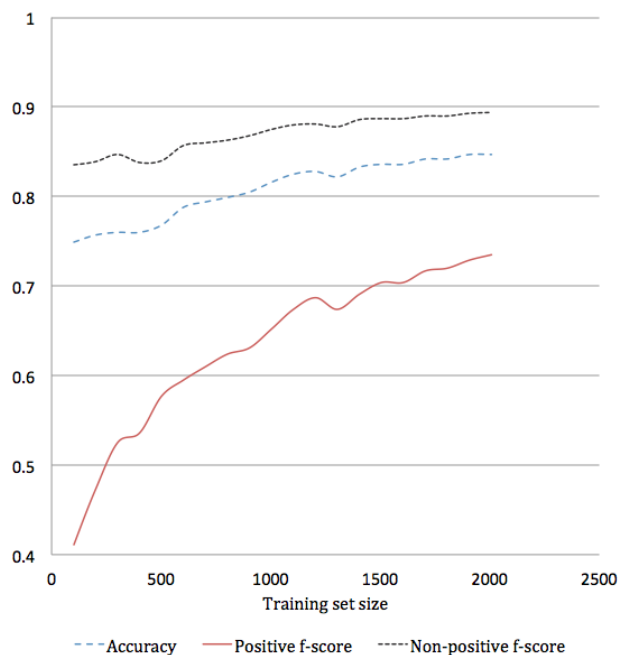


Figure 2: Classification accuracies, and positive and non-positive class f-scores for training sets of various sizes.

positive class, particularly, suffers from the lack of available training data. The increasing gradients for all three curves indicate that if more training data were available, better results could be obtained for both the classes. This is particularly true for the positive class, which is also perhaps the more important class considering our goal of generating bottom-line recommendations for clinical queries. The highest accuracy obtained by our system is 84.7%, which is significantly better than the baseline system for this domain.

To conclude this investigation, we performed manual evaluation to validate the suitability of the polarity classification approach for the generation of bottom-line recommendations. We used the 33 questions from our preliminary analysis for this. We ran 10-fold cross validation on the whole

data set, collected all the sentences associated with these 33 questions, and computed the precision and recall of the automatically identified polarities of the interventions by comparing them with the annotated bottom-line recommendations. The results obtained by the automatic system were: recall - 0.62, precision - 0.82, f-score - 0.71. Understandably, the recall is low due to the small amount of training data available for the positive class, and the f-score is similar to the f-score obtained by the positive class in the polarity classification task.

## 5 Conclusion and Future Work

We presented an approach for automatic, context-sensitive, sentence-level polarity classification for the medical domain. Our analyses on a specialised corpus showed that individual sentence-level polarities agree strongly with the polarities of bottom-line recommendations. We showed that the same sentence can have differing polarities, depending on the context intervention. Therefore, incorporating context information in the form of features can be vital for accurate polarity classification. Our machine learning approach performs significantly better than the baseline system with an accuracy of 84.7%, and an f-score of 0.71 for the bottom-line recommendation prediction task.

Post-classification analyses showed that the most vital aspect for improving performance is the availability of training data. Research tasks specific to a specialised domain, such as the medical domain, can significantly benefit from the presence of more annotated data. Due to the promising results obtained in this paper, and the importance of this task, future research should focus on annotating more data and utilising them for improving classification accuracies. Our future research will also focus on implementing effective strategies for combining the contextual sentence-level polarities to generate bottom-line recommendations.

## References

- Alan R. Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: The metamap program. In *Proceedings of AMIA Annual Symposium*, pages 17–21.
- Yonggang Cao, Feifan Liu, Pippa Simpson, Lamont D. Antieau, Andrew Bennett, James J. Cimino, John W. Ely, and Hong Yu. 2011. AskHermes: An Online Question Answering System for Complex Clinical Questions. *Journal of Biomedical Informatics*, 44(2):277–288.
- Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.
- Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. Evaluation of negation phrases in narrative clinical reports. In *Proceedings the AMIA Annual Symposium*, pages 105–109.
- John W. Ely, Jerome A. Osheroff, Mark H. Ebell, George R. Bergus, Barcey T. Levy, M. Lee Chambliss, and Eric R. Evans. 1999. Analysis of questions asked by family doctors regarding patient care. *BMJ*, 319(7206):358–361, August.
- Su Nam N. Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support Evidence Based Medicine. *BMC bioinformatics*, 12 Suppl 2.
- J Richard Landis and Gary G Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, March.
- Jimmy J. Lin and Dina Demner-Fushman. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103.
- John Lyons. 1981. *Language, Meaning and Context*. Fontana, London.
- Diego Mollá-Aliod and Maria Elena Santiago-Martinez. 2011. Development of a Corpus for Evidence Based Medicine Summarisation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, December.
- Yun Niu, Xiaodan Zhu, Jianhua Li, and Graeme Hirst. 2005. Analysis of polarity information in medical text. In *Proceedings of the AMIA Annual Symposium*, pages 570–574.
- Yun Niu, Xiaodan Zhu, and Graeme Hirst. 2006. Using outcome polarity in sentence extraction for medical question-answering. In *Proceedings of the AMIA Annual Symposium*, pages 599–603.
- 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 of the Association for Computational Linguistics*, pages 271–278.
- Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1):1–135.
- 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 (EMNLP)*.
- Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- Kenji Sagae and Jun’ichi Tsujii. 2007. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 1044–1050.
- Abeed Sarker, Diego Mollá, and Cécile Paris. 2011. Outcome Polarity Identification of Medical Papers. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 105–114, December.
- Abeed Sarker, Diego Mollá, and Cécile Paris. 2012. Towards Two-step Multi-document Summarisation for Evidence Based Medicine: A Quantitative Analysis. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, pages 79–87, Dunedin, New Zealand, December.
- Abeed Sarker, Diego Mollá, and Cécile Paris. 2013. An approach for query-focused text summarisation for evidence based medicine. In Niels Peek, Roque Marn Morales, and Mor Peleg, editors, *Artificial Intelligence in Medicine*, volume 7885 of *Lecture Notes in Computer Science*, pages 295–304. Springer Berlin Heidelberg.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2010. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307.
- Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-02, 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, US. Association for Computational Linguistics.
- Vladimir N. Vapnik. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- 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 (Suppl 11)(S9).
- Theresa Wilson, Janyce Wiebe, and Paul Hoffman. 2009. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics*, 35(3):399–433.

# Clustering Microtext Streams for Event Identification

Jie Yin

Computational Informatics

CSIRO, Australia

Jie.Yin@csiro.au

## Abstract

The popularity of microblogging systems has resulted in a new form of Web data – microtext – which is very different from conventional well-written text. Microtext often has the characteristics of informality, brevity, and varied grammar, which poses new challenges in applying traditional clustering algorithms to analyze microtext. In this paper, we propose a novel two-phase approach for clustering streaming microtext, in particular Twitter messages, into event-based clusters. In the online phase, an incremental process is applied to discover base clusters and maintain detailed summary statistics. Upon demand for any user-specified time horizons, an offline phase is triggered to merge related clusters together. We demonstrate that our proposed approach can achieve better clustering accuracy than state-of-the-art methods.

## Introduction

Microtext is a newly emerging type of Web data which is generated in enormous volumes with the proliferation of online microblogging systems. These systems, such as Twitter and Facebook, provide a light-weight, easy form of communication that enables individuals around the globe to share information and express their opinions in fluid and less formal ways. Microtext streams generated from these sites offer a rich source of real-time information about a wide variety of real-world events, ranging from planned occurrences such as political campaigns or sports games, to unexpected incidents such as earthquakes or terrorist riots. To provide insight into user-generated content broadcast in microtext streams, clustering approaches have demonstrated great potential for

identifying what topics people are talking about and tracking how events unfold over time.

Clustering microtext streams poses a number of new challenges, due to short, noisy and informal nature of microtext [Ellen, 2011]. First, clustering techniques should be scalable to the sheer volume of data generated in microblogging systems. Twitter, for example, generates over 400 million tweets per day in early 2013. Thus, it is crucial to develop efficient clustering algorithms that can handle such massive amounts of streaming data. Second, microtext often has the characteristic of informality, brevity, varied grammar, and free-style. Depending on various personal style or background knowledge, people tend to use different words to convey the same or similar meanings, when writing about a particular event. Therefore, it is highly desirable to design effective clustering algorithms that can discover event-based clusters over time.

To cope with the sparsity and brevity of microtext, different methods have been proposed for microtext clustering in recent years. The majority of previous work has primarily focused on clustering a static collection of short documents [Rangrej et al., 2011, Tsur et al., 2012], or on using surface features to compute pairwise similarity between microtext [Reuter et al., 2011, Li et al., 2012]. However, the challenge of how to effectively cluster microtext in dynamic data streams has not been well addressed.

In this paper, we propose a novel framework for automatically grouping streaming microtext, in particular Twitter messages, into a set of event-based clusters; it intelligently divides the clustering process into an online component which maintains summary statistics, and an offline component which uses these compact statistics to discover event-based clusters. In the online phase, an incremental process is applied to discover base clusters and maintain detailed summary statistics about the clusters. This process can be efficiently

performed for the purpose of online social media monitoring. The generated base clusters serve as an intermediate statistical representation of the stream. Upon request, an offline phase is thereafter utilized to perform more computational analyses which merge similar clusters together in a bottom-up manner within a given time horizon. Experimental results show that our proposed clustering algorithm improve the clustering quality of other state-of-the-art approaches.

## Related Work

This section reviews two primary related research areas: first, short text clustering which deals with very short and informal text; and second, studies that address event identification in social media.

### Short Text Clustering

Although document clustering is well studied in the past decade, clustering very short, noisy and informal text has remained a challenging task. Rosa et al. [2010] studied the problem of clustering tweets into several pre-specified categories. They used hashtags as indicators of topics and argued that the clusters produced by traditional unsupervised methods can often be incoherent from a topical perspective. Rangrej et al. [2011] compared the performance of three document clustering techniques on Twitter data, and found that graph-based approach using affinity propagation performs best in clustering tweets. To cope with the sparsity of tweets, Tsur et al. [2012] constructed a virtual document by concatenating all micro-messages having the same hashtag, and then applied  $k$ -means algorithm to cluster virtual documents. Existing research has primarily focused on clustering a static collection of short text, while the challenge of continuously clustering microtext streams has not been well addressed.

### Event Identification in Social Media

In recent years, identifying events from social media has attracted much attention. Petrović et al. [2012] applied a  $k$ -nearest neighbor approach to detect the first message talking about an event in a stream of Twitter messages, and used locality-sensitive hashing to speed up the computational process. Reuter et al. [2011] formulated the event identification problem as a record linkage task, in which a blocking strategy was used to reduce the number of pairs of documents consid-

ered for computing pairwise similarity. Becker et al. [2011] proposed an incremental clustering approach to group Twitter messages into clusters, which was similar to the method developed for detecting events in streams of text documents [Allan et al., 1998]. This approach determines the assignment of a message based on its similarity to textual centroids of existing clusters. Li et al. [2012] proposed to first detect bursty tweet segments as event segments and then use graph-based clustering to cluster event segments into events. Most of these works have either relied on computing pairwise similarity between static messages, or considered only the textual features of messages. In our work, however, we focus on developing an efficient framework for clustering a continuous stream of microtext, which groups clusters in a single pass and has the flexibility to merge clusters upon demand to identify event-based clusters.

### Microtext Stream Clustering

We aim to design an effective microtext stream clustering algorithm that can meet three requirements: (1) The ability to handle massive volumes of microtext (*i.e.*, tweets) under the one-pass constraint of streaming scenarios; (2) The ability to employ temporal information in the clustering process, because tweets published within a certain time interval are more likely to correspond to the same event in the stream; (3) The ability to merge related clusters together when necessary. To meet these needs, we propose a new clustering framework which works in two phases, *i.e.*, an online discovery phase and an offline cluster merging phase. The basic idea is to carefully balance the computational load between the online component and the offline component. In the online phase, the Twitter stream is processed in a single pass to maintain sufficient summary statistics about the evolving stream. The offline phase provides the flexibility for an analyst to perform queries about clusters and retrieve event-based clusters upon demand over different time horizons.

Below, we detail the two phases in the following two subsections.

#### Online Discovery Phase

The main task of the online phase is to provide a one scan algorithm over the incoming Twitter stream for identifying base clusters, with each cluster consisting of a set of similar tweets. For

this purpose, we design an efficient single-pass clustering algorithm which clusters the stream of tweets in an incremental manner.

To represent textual information of tweets, we employ a traditional vector-space model which uses the bag-of-words representation. A tweet is represented using a vector of words (terms or features), which are weighted using the term frequency (TF) and the inverse document frequency (IDF) [Salton and Buckley, 1988]. Using this model, a tweet represents a data point in  $d$ -dimensional space,  $\mathbf{m}_i = (v_1, v_2, \dots, v_d)$ , where  $d$  is the size of the word vocabulary and  $v_j$  is the TF-IDF weight of  $j^{\text{th}}$  word in tweet  $m_i$ . However, in a dynamic microtext stream, word vocabulary changes and the number of tweets increases over time, making it computationally expensive to recalibrate the inverse document frequency of TF-IDF. Therefore, we resort to using term frequency as the term weight and adopting a sparse matrix representation of tweets to deal with dynamically changing vocabulary in our clustering algorithm.

To discover meaningful clusters, one important factor is defining an effective similarity measure. In our work, we use cosine similarity to measure textual similarity between two tweets, which is defined as

$$\text{sim}_{\text{text}}(\mathbf{m}_i, \mathbf{m}_j) = \frac{\mathbf{m}_i \cdot \mathbf{m}_j}{\|\mathbf{m}_i\| \times \|\mathbf{m}_j\|}, \quad (1)$$

where  $\mathbf{m}_i \cdot \mathbf{m}_j$  indicates the dot product of vectors  $\mathbf{m}_i$  and  $\mathbf{m}_j$ . Besides,  $\|\mathbf{m}_i\|$  and  $\|\mathbf{m}_j\|$  denotes the norm of vectors  $\mathbf{m}_i$  and  $\mathbf{m}_j$ , respectively.

Since real-world events typically span a limited time interval, tweets that largely differ on their publication times are much less likely to belong to the same event. Therefore, in order to cluster tweets into temporally-related groups, we also exploit a time similarity measure defined as

$$\text{sim}_{\text{time}}(\mathbf{m}_i, \mathbf{m}_j) = \exp\left(-\frac{|t_{\mathbf{m}_i} - t_{\mathbf{m}_j}|}{\lambda}\right), \quad (2)$$

which is based inversely on the distance between tweets' publication dates/times.  $|t_{\mathbf{m}_i} - t_{\mathbf{m}_j}|$  indicates the time difference between tweets  $\mathbf{m}_i$  and  $\mathbf{m}_j$ , represented as the number of days, and  $\lambda$  is the number of days of one month, whose value is application dependent. In our case, if  $t_{\mathbf{m}_i}$  and  $t_{\mathbf{m}_j}$  are more than one month apart, we consider time similarity between  $\mathbf{m}_i$  and  $\mathbf{m}_j$  to be very small.

Putting together, our clustering algorithm uses a

combined similarity measure defined as:

$$\text{sim}(\mathbf{m}_i, \mathbf{m}_j) = \text{sim}_{\text{text}}(\mathbf{m}_i, \mathbf{m}_j) \cdot \text{sim}_{\text{time}}(\mathbf{m}_i, \mathbf{m}_j). \quad (3)$$

This similarity measure not only captures the similarity between the textual vectors of tweets, but also penalizes the similarity between tweets if their publication dates/times are far away.

To maintain sufficient information about clusters, we represent each cluster  $C_i$  using a cluster feature vector  $\psi(C_i)$ , defined as follows:

- Textual centroid  $C_i^w$ : which is a vector in which each element represents the average weight of the corresponding words for all tweets in cluster  $C_i$ .
- Time centroid  $C_i^t$ : which is the average publication time of all tweets that form cluster  $C_i$ .
- Cluster size  $|C_i|$ : which is defined as the number of tweets belonging to cluster  $C_i$ .

Now we describe the process of the incremental clustering algorithm. Given a Twitter stream in which the tweets are sorted according to their published times, the algorithm takes the first tweet from the stream, and uses it to form a cluster. As a new tweet  $\mathbf{m}$  arrives, we calculate the similarity between tweet  $\mathbf{m}$  and any existing clusters  $C_i$  as

$$\text{sim}(\mathbf{m}, C_i) = \text{sim}_{\text{text}}(\mathbf{m}, C_i^w) \cdot \text{sim}_{\text{time}}(t_{\mathbf{m}}, C_i^t). \quad (4)$$

Let  $C$  be the cluster that has the maximum similarity with  $\mathbf{m}$ . If  $\text{sim}(\mathbf{m}, C)$  is less than a similarity threshold  $\delta_{\text{sim}}$ , which is to be determined empirically, a new cluster is created to include  $\mathbf{m}$ ; Otherwise, the tweet  $\mathbf{m}$  is assigned to the closest cluster  $C$ . By adjusting the threshold  $\delta_{\text{sim}}$ , we can obtain clusters at different levels of granularity. Once a new tweet  $\mathbf{m}$  is added to cluster  $C_i$ , we update the corresponding cluster representatives  $\psi(C_i)$  using the following equations:

$$\hat{C}_i^w = \frac{C_i^w \times |C_i| + \mathbf{m}}{|C_i| + 1}, \quad (5)$$

$$\hat{C}_i^t = \frac{C_i^t \times |C_i| + t_{\mathbf{m}}}{|C_i| + 1}, \quad (6)$$

$$|\hat{C}_i| = |C_i| + 1. \quad (7)$$

This incremental algorithm is efficient as it considers each tweet at once, and can thus scale to a growing amount of tweets. To further improve efficiency, we maintain a list of active clusters over

time in the online phase. If no more tweets are added to a cluster for a period of time, which is determined based on application needs, the cluster is considered inactive and it is removed from the active list. The algorithm considers only those clusters in the active list as candidates to which a new tweet can be added. The output of the algorithm is a list of clusters  $C_1, \dots, C_H$ , together with their cluster representatives  $\psi(C_1), \dots, \psi(C_H)$ .

### Offline Cluster Merging Phase

The base clusters generated by the online phase serve as an intermediate statistical representation, which can be maintained in an efficient way even for a large volume of tweets. The subsequent offline phase is utilized to merge a list of clusters into event-based clusters. There is no need to process the voluminous microtext stream, but the compactly stored summary statistics of clusters.

For a particular event, since users tend to convey the same or a similar meaning using different words depending upon their own personal style, the online phase would organize the tweets that report the same event, but expressed using different words, into different base clusters. Therefore, we propose to merge together the clusters that are related with respect to the same event in the offline phase. Concretely, we calculate a cluster merge criterion,  $link(C_i, C_j) = sim_{text}(C_i^w, C_j^w) \cdot sim_{time}(C_i^t, C_j^t)$ , which captures the inter-similarity between two clusters  $C_i$  and  $C_j$ . The principle is to merge a pair of clusters that have a larger inter-cluster similarity. When two clusters are merged, we merge a smaller cluster into the larger one and in this way, larger clusters are retained which can better represent significant events of interest.

The offline clustering phase provides the flexibility to query the clustering results at any time horizon. Given a list of clusters generated during the online phase, we consider iteratively merging two clusters  $C_{j^*}$  and  $C_{i^*}$  such that  $link(C_{i^*}, C_{j^*})$  is maximized. Accordingly, cluster representatives for cluster  $C_{i^*}$  are updated as follows:

$$\hat{C}_{i^*}^w = \frac{C_{i^*}^w \times |C_{i^*}| + C_{j^*}^w \times |C_{j^*}|}{|C_{i^*}| + |C_{j^*}|}, \quad (8)$$

$$\hat{C}_{i^*}^t = \frac{C_{i^*}^t \times |C_{i^*}| + C_{j^*}^t \times |C_{j^*}|}{|C_{i^*}| + |C_{j^*}|}, \quad (9)$$

$$|\hat{C}_{i^*}| = |C_{i^*}| + |C_{j^*}|. \quad (10)$$

To determine an optimal number of clusters,

we use the notion of *separation* to measure the clustering quality, which is defined as the average inter-cluster similarity over all the clusters, that is,  $S(k) = \frac{1}{N(N-1)} \sum_i \sum_j link(C_i, C_j)$ , where  $C_1, \dots, C_N$  are the clusters obtained at step  $k$ . The smaller value this metric has, the better clusters are separated from each other. Based on this metric, we design a criterion to decide whether or not to stop the merging process. At each step  $k$ , given two candidate clusters to be merged, we compute a validation index as

$$\Delta_k = \frac{S(k+1) - S(k)}{S(k)}, \quad (11)$$

which represents the relative change in inter-cluster similarity after a merge is made. If  $\Delta_k < 0$ , that means a cluster merge can improve the separation of clusters. We thus proceed with merging the two clusters. Otherwise, if  $\Delta_k \geq 0$ , we stop the cluster merging process. In this way, the optimal number of clusters can be automatically determined during the cluster merging process.

## Experiments

We carry out experiments to evaluate the effectiveness of our proposed algorithm, and compare its performance with other baseline methods.

### Dataset

The dataset we used is an annotated corpus of tweets collected from the beginning of July 2011 to September 2011 [Petrović et al., 2012]. The corpus was distributed as a set of tweet IDs, together with their annotations. We re-retrieved the tweets using Twitter search API<sup>1</sup> and obtained a set of 2,633 tweets. Each tweet was annotated as one out of 27 events, which cover a variety of real-world events, such as London riots, terrorist attacks in Norway, Earthquake in Virginia, and NASA's announcement about discovery of water on Mars. The annotations are used as the ground truth for evaluating the clustering algorithms.

We preprocessed the tweets by removing stopwords, user mentions (@username), and embedded links, because such elements in tweets may not be useful for indicating the topics. We compiled a list of stopwords that specifically suited Twitter content. It includes formal English stopwords such as *is*, *am*, informal English stopwords

<sup>1</sup><https://dev.twitter.com/docs/using-search>

such as *gonna*, *arent*, and Twitter specific stop-words such as *RT* that indicates a retweet. We also performed a shallow lexical normalization on tweets and stemmed words using Porter Stemmer. For lexical normalization, we only considered words that were emphasized by repeating one or more letters. If a letter was repeated more than three times, it was normalized to one instance of that letter. For example, the word *crazyyyyy* was turned to *crazy*.

For our clustering task, we constructed a Twitter stream by sorting all tweets according to their publication times. The stream was taken as input to the clustering algorithms. For each tweet, we mainly used bag-of-words and specific hashtags (words preceded with a # sign) as features to construct a vector model.

### Baselines

Our proposed algorithm is referred to as **MSC** (Microtext Stream Clustering). For comparison, we use two other methods as baselines:

- **IC**: which is a standard incremental clustering algorithm adopted by Becker et al. [2011]. It determines the assignment of a message solely based on its similarity to the textual centroids of existing clusters.
- **IC-Time**: which differs from our proposed algorithm in that it only uses the first on-line phase to discover clusters. By comparing with this baseline, we show how much gain in clustering quality can be achieved with the offline cluster merging.

In our experiments, we set parameter  $\lambda$  in Eq.(2) to be 30. In addition, we set the similarity threshold  $\delta_{sim} = 0.2$  for all the algorithms.

### Evaluation Metrics

Let  $\mathcal{C} = \{C_1, \dots, C_K\}$  denote the clustering result produced by one clustering algorithm, and  $\mathcal{G} = \{G_1, \dots, G_L\}$  denote the desired ground truth. We use two evaluation metrics: F-measure [Yin and Yang, 2005] and normalized mutual information (NMI) [Strehl and Ghosh, 2003], to validate the effectiveness of the clustering algorithms. we observe that the results are strongly correlated on the two metrics.

### Experimental Results

We first performed experiments to evaluate the performance of three clustering algorithms on the

entire stream. Since hashtags are considered as good indicators of topics in the tweets, we investigated two different ways of using hashtags as features: first, considering hashtags in the same way as words, and second, removing the # symbol and treating hashtags as normal words. Table 1 reports the clustering accuracy using the three algorithms on the two settings.

		F-measure	NMI
<i>Hashtags</i>	IC	0.892	0.897
	IC-Time	0.905	0.907
	MSC	<b>0.958</b>	<b>0.955</b>
<i>Hashtags without #</i>	IC	0.899	0.907
	IC-Time	0.910	0.913
	MSC	<b>0.966</b>	<b>0.962</b>

Table 1: Comparison of clustering algorithms on F-measure and NMI metrics

The top part of the table compares the performance of the three algorithms using bag-of-words and original hashtags as features. We can see that, our proposed MSC algorithm is superior to the other two baselines, while IC-time performs slightly better than IC. This is because, IC only relies on the cosine similarity between textual features of tweets to form clusters, while IC-Time enforces a time constraint in the similarity measure to reflect the time locality of events, which thus leads to better clustering accuracy. By explicitly merging related clusters, MSC achieves the highest accuracy on both two metrics.

The bottom part of the table shows the clustering results by removing the # symbol and treating hashtags as normal words. We can observe that, this improves the clustering accuracy for all three algorithms. We believe that this improvement is because removing the # symbol contributes to increasing the term frequency of the same topic word in the tweets. It thus translates to yielding better clustering accuracy. This can be illustrated using the examples as follows.

*Bold move as Google Buys Motorola for 12.5 Billion, and paid cash #google #motorola.*

*5.8 earthquake happened in Virginia just moments ago. #Earthquake #Virginia.*



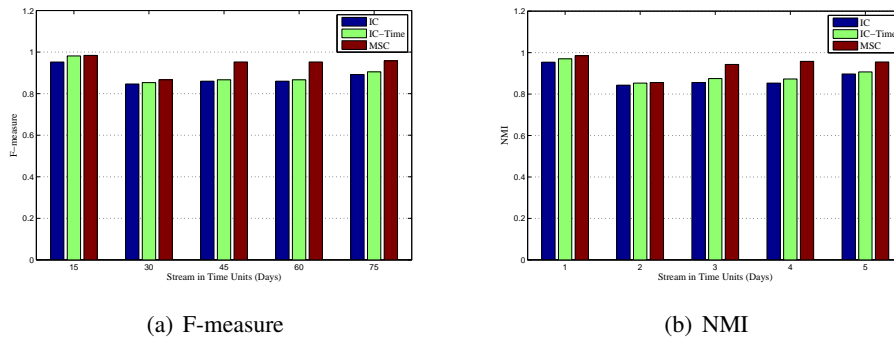


Figure 1: Clustering accuracy over different time horizons

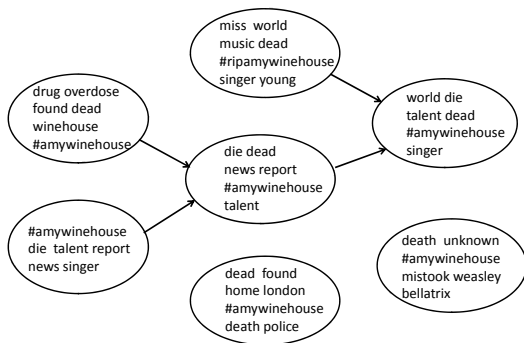


Figure 2: Illustration of the cluster merge process

If we remove the # symbol, hashtags *#google* and *#motorola* are turned into words *google* and *motorola*, in the first tweet, and *#Earthquake* and *#Virginia* are into *Earthquake* and *Virginia*, in the second tweet. In both cases, this increases the term frequencies of the topic words or main entities of events, thus highlighting their contributions to forming the clusters.

To better understand how our MSC algorithm performs cluster merges, Figure 2 illustrates the cluster merging process for the topic talking about the death of Amy Winehouse<sup>2</sup>. There are seven clusters generated from the online phase, each of which is represented using top-ranked keywords in the figure. In the offline phase, the clusters are merged based on their similarity and relatedness in a bottom-up manner, and finally three clusters remain after two rounds of cluster merges.

The other important feature of our proposed MSC algorithm is that it can merge related clusters upon demand for any user-specified horizon. Therefore, we carried out experiments to compare the clustering quality of the three algorithms at dif-

<sup>2</sup>[http://en.wikipedia.org/wiki/Amy\\_Winehouse](http://en.wikipedia.org/wiki/Amy_Winehouse)

ferent time horizons. Figure 1 shows the clustering accuracy with respect to F-measure and NMI at different time units in the stream. We can see that, our proposed MSC algorithm consistently outperforms the other two baselines over time. This indicates that, MSC has the ability to retain sufficient statistics required for effective cluster merging in the offline phase.

## Conclusions and Future Work

In this paper, we proposed a new approach for clustering microtext streams into event-based clusters. Our proposed approach intelligently divides the clustering process into an online component which maintains summary statistics, and an offline component which uses these compact statistics to discover event-based clusters. Therefore, it has the advantage of processing and scaling to large volumes of microtext streams. Experiments and comparisons demonstrated that our proposed approach achieves better clustering accuracy than state-of-the-art methods, and merging similar clusters can improve the performance of short text clustering.

This work can be extended in several directions. We will further evaluate the effectiveness of our clustering algorithm in the ESA (Emergency Situation Awareness) system [Yin et al., 2012] in larger-scale datasets. In particular, we will test its performance together with the burst detection module for identifying significant event-based clusters from the real-time Twitter stream. Moreover, since short, informal microtext has high degree of lexical variations, we will explore paragraphing techniques to uncover hidden semantic relatedness between microtext. Such information can be leveraged to group clusters that talk about the same event, but expressed using different words, and thus improve the clustering quality.

## References

- J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–45, Melbourne, Australia, 1998.
- H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, pages 438–441, Barcelona, Catalonia, Spain, 2011.
- J. Ellen. All about microtext: A working definition and a survey of current microtext research within artificial intelligence and natural language processing. In *Proceedings of the Third International Conference on Agents and Artificial Intelligence*, pages 329–336, Rome, Italy, 2011.
- C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 155–164, Maui, HI, USA, 2012.
- S. Petrović, M. Osborne, and V. Lavrenko. Using paraphrases for improving first story detection in news and Twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–346, Montreal, Canada, 2012.
- A. Rangrej, S. Kulkarni, and A.V. Tendulkar. Comparative study of clustering techniques for short text documents. In *Proceedings of the International World Wide Web Conference*, pages 111–112, Hyderabad, India, 2011.
- T. Reuter, P. Cimiano, L. Drumond, K. Buza, and L. Schmidt-Thieme. Scalable event-based clustering of social media via record linkage techniques. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, pages 313–320, Barcelona, Catalonia, Spain, 2011.
- K.D. Rosa, R. Shah, B. Lin, A. Gershman, and R. Frederking. Topical clustering of tweets. In *Proceedings of the SIGIR Workshop on Social Web Search and Mining*, Beijing, China, 2010.
- G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
- A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003.
- O. Tsur, A. Littman, and A. Rappoport. Scalable multi stage clustering of tagged micro-messages. In *Proceedings of the 21st World Wide Web Conference*, pages 621–622, Lyon, France, 2012.
- J. Yin and Q. Yang. Integrating hidden Markov models and spectral analysis for sensory time series clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 506–513, Houston, Texas, USA, 2005.
- J. Yin, S. Karimi, B. Robinson, and M. Cameron. Esa: Emergency situation awareness via microbloggers. In *Proceedings of the Twenty-First ACM International Conference on Information and Knowledge Management*, pages 2701–2703, Maui, Hawaii, USA, 2012.

# Automatic Corpora Construction for Text Classification

Dandan Wang, Qingcai Chen, Xiaolong Wang, Bingyang Yu  
Key Laboratory of Network Oriented Intelligent Computation  
Harbin Institute of Technology Shenzhen Graduate School, China  
{wangdandanhit, qingcai.chen}@gmail.com  
wangxl@insun.hit.edu.cn  
bingyang.yu@hotmail.com

## Abstract

Since the machines become more and more intelligent, it is reasonable to expect the automatic construction of text classifiers by given just the objective categories. As trade-off solutions, existing researches usually provide additional information to the category terms to enhance the performance of a classifier. Unique from them, in this paper, we construct the standard corpora from the web by just providing text categories. Since there are millions of manually constructed websites, it is hopeful to find out proper text categorization (TC) knowledge. So we directly go to the web and use the hierarchies implied in navigation bars to extract and verify TC resources. By addressing the issues of navigation bar recognition and text filtering, the corpora are constructed for given text categories and the classifiers are trained based on them. We conduct our experiments on the large scale of webpages collected from the 500 top English websites on Alexa. The Open Directory Project (ODP) is used as testing corpus. Experimental results show that, being compared with the classifier based on manually labeled corpus, the classifier trained on auto-constructed corpora reaches comparable performance for the categories that are well covered by the training corpus.

## 1 Introduction

As one of the key techniques in web information processing, text classification has been studied for a long time (Aas and Eikvil, 1999; Wang and Li, 2011; Yang and Pedersen, 1997). A growing number of machine learning techniques have been applied to text classification and some of them

have proven to be successful (Miao and Kamel, 2011; Sebastiani, 2002). In the machine learning approach, the learning process is an instance of supervised or semi-supervised learning because classifier can be built automatically by learning from adequately pre-labeled training documents and then classified unseen documents (Feldman and Sanger, 2007). However, the task of manually labeling a large amount of documents is time-consuming and even impractical. Given a general classification task, people usually construct training data in two ways. One is augmenting a small number of labeled documents with large amounts of unlabeled ones to guide the learning model iteratively, so that the new classifier can label the unlabeled documents (Jiang, 2009; Nigam et al., 2000). In such studies, the bootstrapping technique is often used to label the unlabeled documents and refine the initial classifier (Gliozzo et al., 2009; Ko and Seo, 2009). The other is collecting training corpora from the Web. Such works use the class name and its associated terms to collect training corpora iteratively (Huang et al., 2005). Cheng (2009) and Day et al. (2009) firstly sample the Web with a set of given class names, and then query the keywords manually populated from each class by search engines for retrieving quality training documents. Huang et al. (2004) proposed a LiveClassifier system which also makes use of search engines for automatically constructing training classifier.

Though reached encouraging performance, above methods have some limitations in organizing training corpora. For the first method, although some algorithms just use a small set of labeled documents, which still require much time and effort for complicated categories (Chen et al., 2009); And the second method depends on several external resources, which greatly limits its flexibility and reliability. For example, manually given keywords or terms for a class are easily affected by

different persons; different search engines may also bring different results with various type of noises contained in search results (Huang et al., 2004).

Inspired by these issues, we design a new system to automatically acquire training corpora. Given a class hierarchy, our basic idea is to collect corpora merely based on class names. Firstly, we crawl the webpages starting with several selected websites and identify the navigation bars of these websites. Then each navigational item in the navigational bars is matched with the class names. The valid subpages from a navigational item are labeled with the matched class name. After extracting contents from these subpages, the initial candidate corpora are constructed. Finally clustering algorithm is used to remove noises from the corpora. In latter parts of paper, we denote the automatic constructed corpora as ACC.

The main contributions of this paper are:

(1) An automatic system for constructing classification corpora is built. It is a new way to collect large-scale, high quality corpora; moreover, it is completely adaptive to any kind of class hierarchy;

(2) To improve the ACC quality, text clustering based automatic noise filtering approaches are proposed and analyzed;

(3) The proposed system and methods are evaluated on large scale standard corpora and encouraging results are reached.

The remainder of this paper is organized as follows. Section 2 described the architecture of the automatic corpora construction system; Section 3 and 4 present experimental settings and results respectively. The paper is closed with conclusion and future work in section 5.

## 2 Automatic Corpora Construction

In this paper, we propose a novel system that can automatically acquire effective training data through web mining. The architecture of the system is given in Fig 1. It is composed of four modules: data collection, navigational processing, candidate corpora construction and corpora denoising. The data collection module crawls webpages from the given URL seeds. The navigational processing module is to extract the navigational bars from downloaded web pages, and to make category judgments for each navigational item. The candidate corpora construction module is to get the candidate corpora by performing content extraction for the valid links from the navigational item-

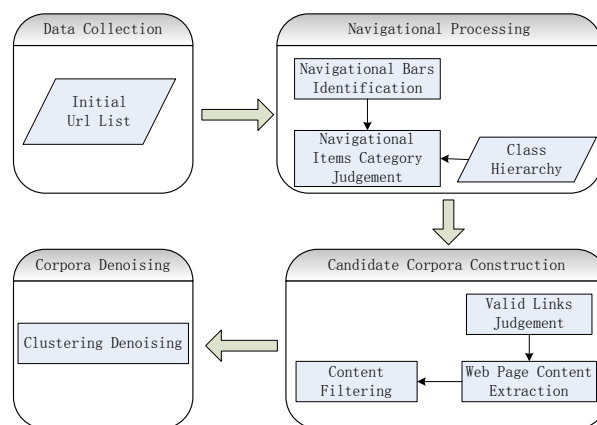


Figure 1: Architecture of the ACC System.

s. The corpora denoising module makes clustering for all texts in the candidate corpora and removes the noisy web pages for each category.

### 2.1 Navigational Bars Processing

Generally, the navigational bars locate in one block or several blocks of a web page. Thus it is necessary to split the web page into blocks for identifying the navigational bars. In this paper, we use the approach proposed by (Keller and Nussbaumer, 2012) for segmentation and simplify it by three rules as follows.

- (1) Remove leaf nodes which are not link nodes;
- (2) If the node is the only child of its parent node, delete the parent node and directly connect the node with its ancestor node;
- (3) If the node has two children nodes, and the first child node is the link node, while the other is not, then delete the node and connect the two children nodes with its ancestor node.

In latter parts of this paper, the set of blocks after segmentation is denoted as PageSec.

#### 2.1.1 Navigational Bars Identification

The approaches for navigational bars identification can be divided into two categories, i.e., rule-based and graph-based filtering respectively. In this paper, we compared two different approaches for the navigational bars identification.

**Rule-based Identification:** Firstly, we conduct filtering for each block in PageSec. The filtering conditions include link type, link uniqueness etc. Then a ranking formula is constructed to calculate the score for each blocks. The features used in the formula are listed as follows.

- (1) **Consistent degree of link depth between the anchor texts within block:** Each anchor text

within the block will point to a link. The depth of a link is represented by its slashes' number. The consistent degree of link depth between the anchor text within block is calculated by formula (1).

$$Dep(Sec) = \frac{-\sum_{i=1}^n (P(AcD_i) \ln(P(AcD_i)))}{\ln(n)} \quad (1)$$

Where  $n$ : The number of anchors with different depth within the block.

$P(AcD_i)$ : The proportion of anchors with the  $i$ th anchor depth within the block.

**(2) Consistent degree of word number between the anchor texts within block:** Generally, the word number of each navigational item is consistent. The more tidy appearance of the items, the more likely to belong to the same navigational bar. The consistent degree of word number between the anchor texts within block is calculated in the same way with the consistent degree of link depth.

**(3) The proportion of the remaining anchor texts within block.**

Integrating the three features above, we construct a linear weighted summation formula to rank for the blocks. Finally, the top-ranking blocks with score bigger than a threshold are added to the candidate of the navigational bars.

**Graph-based Identification:** Firstly, we construct the relationship graph of links and find the maximum complete subgraph by the Bron-Kerbosch algorithm. Then we use a discriminant algorithm to extract the final navigational bars.

**(1) Construction of Links Relationship Graph:** Each web page is represented with a node, if page A has a link pointing to page B, a directed edge from A to B is generated. We deleted single directional edges and preserved bi-directional edges; moreover, the directed graph is transformed to be undirected for simplification. In this paper, the Bron-Kerbosch algorithm is used to get the maximum complete subgraph.

**(2) Extraction of Navigational Bars:** For extracting navigational bars, we need to take advantage of maximum complete subgraphs to conduct filtering on the block structure of the web page. The pseudo code is listed in algorithm1. In addition, as the vertex number increases to a value, the running time of searching for the maximum subgraphs becomes unacceptable. Thus we apply an approximate algorithm to extract the navigational

bars when the vertex number is more than 100.

## 2.1.2 Navigational Items Category Judgment

After identifying the navigational bars, we need to match the navigational items with each class of the given class hierarchy. The cosine similarity in the vector space model is used for calculating the matching degree. Moreover, in order to get a better result, we apply stemming for the navigational items and word expansion for the class names. Given a navigational item, we firstly calculate its similarities with all classes and sort those classes in descending order of the similarities. If the maximum similarity in the rank is unique and greater than a given threshold, label the navigational item with the corresponding class. Otherwise, we use the URL information to make further judgment.

---

### Algorithm1: Navigational bars extraction

**Input:** Maximum complete subgraph MCSQueue  
Block set of homepage PageSec

**Output:** Candidate navigational bars CandNav

Step1: Label all the elements in MCSQueue with unprocessed.

Step2: Select an unprocessed subgraph SGraph from MCSQueue. If all processed, turn to step 4.

Step3: Filter on all the elements in PageSec, remove the elements not in SGraph and save the result in CandNav, turn to step 2.

Step4: Sort all blocks in CandNav from more to less by the number of elements.

Step5: Traverse CandNav from the beginning and delete Sec if current block contains all the elements of block Sec which is at the position behind.

Step6: End

---

## 2.2 Candidate Corpora Construction

### 2.2.1 Valid Links Judgment

A navigational item within the navigational bar usually points to a hub web page that is mainly composed of a set of links, which makes it difficult to extract relevant web pages for a given topic. In general, the links that point to external websites are treated as invalid links and can be directly filtered. Since some kind of invalid links, such as the *Login*, *Sitemap* etc, occur many times in the website and the times of category assignment to the invalid links are significantly more than the valid links. In this paper, we regard the links with the times of category assignment more than a threshold as invalid links. Since there are multiple nav-

igational paths that can lead to a web page, each valid link may have multiple category labels and the voting strategy is used to label the valid link.

### 2.2.2 Web Page Content Extraction

Extracting content from webpages has been researched for decades and numerous methods have been proposed. Tim proposes a method to extract content text from diverse web pages by using the HTML documents tag ratios (Weninger et al., 2010). Sun presents Content Extraction via Text Density (CETD) to extract content and also proposes a method called DensitySum to extract integral content. Moreover, CETD-DS has shown that it is an effective and robust content extraction algorithm (Sun et al., 2011). In this paper, we apply Tim’s tag ratio to extract content.

**Tag Ratio:** Tim’s tag ratio is the ratio of HTML tags characters and Non-HTML tags characters at each row of the HTML source code. It can be described by formula (2) as follows:

$$TagRatio_i = \frac{NonTagChars_i}{TagNum_i} \quad (2)$$

Where  $NonTagChars_i$ : The number of Non-HTML tags characters at the  $i$ th row.

$TagNum_i$ : The number of HTML tags at the  $i$ th row. For the hyperlink tag, multiply it by 2.

Since the comments, scripts and CSS tags do not contain the text content, we remove them from the HTML code while calculating the tag ratio. For the tag ratio, we apply the standard gaussian approach to make smoothing. Firstly we construct a gaussian kernel (Keerthi and Lin, 2003) with radius of 1 and variance of 2 by formula (3).

$$k_i = \sum_{j=-[\sigma]}^{[\sigma]} e^{-\frac{j^2}{2\sigma^2}}, 0 \leq i \leq 2[\sigma] \quad (3)$$

After normalization, we get (4):

$$k'_i = \frac{k_i}{\sum_{j=0}^{2[\sigma]} k_j}, 0 \leq i \leq 2[\sigma] \quad (4)$$

Then we make the convolution operation with this filter and tag ratio to get the smoothing tag ratio.

**Content Extraction:** Most of the methods for content extraction try to use different threshold selection strategies for different features, but the generalization performance of these algorithms are still not satisfactory. Therefore, we adopt the K-means clustering with two-dimensional features

to get the final content. The first dimensional feature is the smoothing tag ratio, the second one is the approximate derivative of the tag ratio.

### 2.3 Corpora Denoising

Ideally, web pages with the same category label in the candidate corpora belong to the same topic. However, because of the different website authority and management level, some web pages which do not belong to the category topic are also classified into this category. In addition, some irrelevant web pages are preserved owing to the weakness of the valid links judgment. These noisy web pages greatly reduce the quality of the candidate corpora. Therefore, we apply K-means clustering algorithms to conduct corpora denoising. After clustering, the large clusters are preserved while the smaller ones are removed as the noises.

## 3 Experimental Settings

### 3.1 Databases

**Self-collected Data:** A universal crawler is implemented for the automatic corpora construction task. In addition to the general crawler task, it records the jumping information between web pages for locating the target web pages and the orders of visiting each page in the crawling process. The initial seed links of the crawler are collected from the Alexa<sup>1</sup> top 500 websites of each category on May 26, 2012. After removing the duplicated websites, there are 5593 ones left. Take them as the starting links and perform crawling, finally we crawled a total of 783035 web pages.

**ODP (Open Directory Project):** ODP is one of the largest corpora for web page classification. We download the ODP corpus on April 22, 2012. There are totally 4066266 web page links in which 2144930 web page links are downloaded.

**Dataset Split:** In our experiments, both the automatic constructed data and ODP are randomly split into three near equal scale of subsets with two parts for training and the remaining for testing.

### 3.2 Experimental Settings

**Preprocessing:** The class hierarchy of ODP is used for the automatic corpora construction. Libxml2 is used to parse the web pages into the DOM trees. For the web pages that cannot be directly parsed, tidy is applied to correct the syntax mistakes. The Porter algorithm (Jongejan and

<sup>1</sup>Alexa:<http://www.alexa.com/>

Dalianis, 2009) is used for stemming. Generally, there are two kind of ways to make word expansion. One is based on WordNet (Fellbaum, 1998) and the other is to obtain the description words for categories by search engine or other external resources. In this paper, the description words for each category come from its secondary classification keywords under the ODP class hierarchy.

**Classification and Clustering:** Expected Cross Entropy is selected for feature selection. 8000 dimensional features are selected out. LibSVM<sup>2</sup> is used for classification where the linear kernel and default settings are applied. K-means is used for clustering where K is set to 8.

## 4 Experimental Results

### 4.1 Performance of ACC system

**Navigational Bars Identification:** To evaluate the performance of navigational bars identification, we randomly select five websites from the initial seed links for human label. The results of rule-based and graph-based navigational bars identification are shown in Table 1.  $N_{nb}$  denotes the number of navigational bars.  $A_{nr}$  and  $R_{nr}$  denote the accuracy and recall of rule-based identification approach.  $A_{ng}$  and  $R_{ng}$  denote the accuracy and recall of graph-based identification approach. We can see that both methods reach high accuracy and relatively lower recall without significant difference. The reason is that some external links are filtered while some navigational bars are composed of items from different domains.

**Navigational Items Category Judgment:** We randomly sampled 100 websites for human label from the initial seeds of the crawler. The performance of navigational items category judgment is demonstrated in Table 2. Since the accuracy of category judgment is very high, a navigational identification method with higher recall is selected to increase the size of the final corpora.

**Web Page Content Extraction:** We evaluate the performance of web page content extraction on seven standard corpora: CleanEval-Eng, NY Times, Yahoo!, Wikipedia, BBC, Arts Technica and Chaos. Table 3 shows that the web page content extraction algorithm performs well on accuracy for all the seven corpora, but the recall is not satisfactory on corpora like Wikipedia and Yahoo!

Website	$N_{nb}$	$A_{nr}(\%)$	$A_{ng}(\%)$	$R_{nr}(\%)$	$R_{ng}(\%)$
CNN	49	65	81.3	53.1	53.1
Yahoo!	41	36.2	52.9	51.2	22.0
EatingWell	81	87.2	78.5	43.6	79.5
Adobe	39	97.5	65.5	36.4	17.8
Cornell	38	100	51.6	65.8	42.1

Table 1: Performance of navigational bars identification

Category	Judgement Accuracy
Arts	31%
Business	100%
Computers	93%
Games	85%
Health	91%
News	88%
Science	100%
Shopping	83%
Society	87%
Sports	98%

Table 2: Category judgment performance of navigational items

Corpora	Accuracy	Recall	F1
CleanEval-Eng	85.91%	75.42%	80.32%
NY Times	84.24%	84.90%	84.57%
Yahoo!	95.40%	66.29%	78.22%
Wikipedia	98.82%	34.82%	51.50%
BBC	93.74%	83.55%	88.35%
Arts Technica	96.23%	92.16%	94.15%
Chaos	86.08%	94.47%	90.08%

Table 3: Performance of web page content extraction on 7 corpora

**Corpora Accuracy:** In Table 4, we demonstrate the performance of ACC on each category.  $N_p$  denotes the number of crawled web pages.  $A$  denotes the accuracy before clustering and  $A_c$  denotes the accuracy after clustering. This table shows that the macro-average accuracy can reach 68.18%. Furthermore, the applying of simple text clustering method contributes up to 2.73% accuracy performance gains, which demonstrates the effectiveness of our denoising method.

### 4.2 Performance of Classification

**Size of ACC and ODP:** The number of crawled web pages and left web pages after content extrac-

<sup>2</sup>Liblinear:<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

tion in ACC and ODP denoted by  $N_p$  and  $N_{cp}$  respectively are given in Table 5. From this table we can see that only a little part of crawled web pages in both data sets contain effective contents. It is because that most of the web pages in ODP are the homepages of websites, and many web pages mainly contain pictures rather than effective contents in ACC.

**Classification Results of ACC and ODP:** In Table 6, we show the overall SVM classification performance of each category on ODP and ACC. The macro-F1 is approximately 72.3% and 79.8% respectively. By comparing the classification results of ODP and ACC on each category, we can see that the performance of ACC is better than ODP on accuracy and recall.

Category	$N_p$	A	$A_c$
Arts	32773	41%	40%
Business	10477	86%	85%
Computers	2752	60%	65%
Games	20280	73%	76%
Health	7875	52%	59%
News	33850	70%	72%
Rigional	2256	70%	74%
Science	2358	71%	70%
Shopping	7820	57%	64%
Society	7381	77%	81%
Sports	14769	63%	64%
<b>Average</b>	—	<b>65.45%</b>	<b>68.18%</b>

Table 4: Performance of ACC on each category

Category	$N_p$		$N_{cp}$	
	ACC	ODP	ACC	ODP
Arts	32773	206887	1327	5043
Business	10477	208069	1167	4641
Computers	2752	95136	55	1487
Games	20280	44800	357	876
Health	7875	52230	252	977
News	33850	7438	1179	376
Science	2358	91841	136	2735
Shopping	7820	71070	103	853
Society	7381	161806	278	4304
Sports	14769	74317	742	1985

Table 5: Size of ACC and ODP for each category

But it is clear that this can not prove ACC is superior to ODP. Finally we train a classifier with all the automatic constructed corpora and use 1/3 proportion of ODP for testing. The results are shown

Class	Accuracy(%)		Recall(%)		F1(%)	
	ODP	ACC	ODP	ACC	ODP	ACC
Arts	74.1	71.6	84.2	77.3	78.8	74.3
Busi	60.9	76.7	65.4	81.9	63.1	79.2
Comp	73.6	79.5	68.5	77.2	71.0	78.3
Games	81.1	79.7	85.0	79.1	83.0	79.4
Health	68.7	87.3	65.9	86.3	67.3	86.8
News	84.6	83.6	83.2	78.7	83.9	81.1
Science	74.1	78.9	70.4	83.2	72.2	81.0
Shop	47.2	86.3	39.6	88	43.1	87.1
Society	72.2	82.9	76.1	56.9	74.1	67.5
Sports	86.8	82.7	87.9	84.4	87.3	83.5

Table 6: SVM classification performance of each category on ODP and ACC

in Table 7. The results show that the classification performance is relatively poor. The main reason is that the coverage of ACC is inadequate, which leads to a big distribution difference between the training and testing data. The factors that influence the coverage of corpora include the size of initial URL seeds, the crawling depth, the recall of navigational bars etc.

Category	Accuracy	Recall	F1
Arts	63.2%	55.5%	59.1%
Business	34.3%	54.6%	42.1%
Computers	57.7%	61.9%	59.7%
Games	37.4%	77.3%	50.4%
Health	22.4%	81.4%	35.1%
News	21.4%	68.9%	32.7%
Science	40.8%	54.5%	46.7%
Shopping	37.0%	17.4%	23.7%
Society	35.2%	13.5%	13.5%
Sports	53.2%	78.7%	63.5%

Table 7: SVM Cross-Test Result

## 5 Conclusion

In this paper, we proposed a new automatic approach which makes use of web resources to construct the corpora. An automatic system for constructing classification corpora is built. Experiments conducted on ACC and ODP show that the automatic corpora construction approach is effective, although the cross-test result is not satisfactory. Future research will focus on solving the coverage problem of ACC.



## Acknowledgment

This work is supported in part by National Natural Science Foundation of China (No.61173075 and No.61272383).

## References

- Alfio Gliozzo, Carlo Strapparava and Ido Dagan. 2009. *Improving text categorization bootstrapping via unsupervised learning*. ACM Transactions on Speech and Language Processing.
- Bart Jongejan and Hercules Dalianis. 2009. *Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike*. 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, pp.145–153.
- Chen-Ming Hung and Lee-Feng Chien. 1999. *Text classification using web corpora and em algorithms*. The Asia Information Retrieval Symposium, pp.12–23.
- Chien-Chung Huang, Kuan-Ming Lin and Lee-Feng Chien. 2005. *Automatic training corpora acquisition through web mining*. IEEE/WIC/ACM Conference on Web Intelligence.
- Chien-Chung Huang, Shui-Lung Chuang and Lee-Feng Chien. 2004. *Liveclassifier: creating hierarchical text classifier through web corpora*. ACM Transactions on Speech and Language Processing.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Eric P. Jiang. 2009. *Semi-supervised text classification using rbf networks*. International Development Association, pp.95–106.
- Fabrizio Sebastiani. 2002. *Machine learning in automated text categorization*. CM Computing Surveys, 34(1):1–47.
- Fei Sun, Dandan Song and Lejian Liao. 2011. *Dom based content extraction via text density*. SIGIR, pp.245–254.
- Jiaxun Wang and Chunping Li. 2011. *An iterative voting method based on word density for text classification*. WIMS.
- Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun and Tom mitchell. 2000. *Text classification from labeled and unlabeled documents using em*. Machine Learning, 39(2-3):103–134.
- Kjersti Aas and Line Eikvil. 1999. *A Survey*. Text Categorization.
- Matthias Keller and Martin Nussbaumer. 2012. *MenuMiner: revealing the information architecture of large web sites by analyzing maximal cliques*. Proceedings of the 21st international conference companion on World Wide Web, ACM: Lyon, France.
- Pu-Jen Cheng. 2009. *Web-based unsupervised text classification*.
- Ronen Feldman and James Sanger. 2007. *Advanced Approaches in Analyzing Unstructured Data*. The Text Mining Handbook, Cambridge University.
- Si Chen, Gongde Guo and Lifei Chen. 2009. *Semi-supervised classification based on clustering ensembles*. Artificial Intelligence and Computational Intelligence.
- S. Sathiya Keerthi and Chih-Jen Lin. 2003. *Asymptotic behaviours of support vector machines with gaussian kernel*. MIT Press, Cambridge University.
- Tim Weninger, William H. Hsu and Jiawei Han. 2010. *content extraction via tag ratios*. WWW, pp.971–980.
- Wei-Yen Day, Chun-Yi Chi, Ruey-Cheng Chen, Pu-Jen Cheng and Pei-Sen Liu. 2009. *Web mining for unsupervised classification*. pp.53–67.
- Yiming Yang and Jan O. Pedersen. 1997. *A comparative study on feature selection in text categorization*. International Conference on Machine Learning, pp.412–420.
- Youngjoong Ko and Jungyun Seo. 2009. *Text classification from unlabeled documents with bootstrapping and feature projection techniques*. Information Processing and Management, 45(1):70–83.
- Yun-Qian Miao and Mohamed Kamel. 2011. *Pairwise optimized rocchio algorithm for text categorization*. Pattern Recognition Letters, 33(2):375–382.

# Learning to Generate Diversified Query Interpretations using Biconvex Optimization

**Ramakrishna B Bairi**

IITB-Monash Research Academy  
IIT Bombay  
Mumbai, India, 400076  
bairi@cse.iitb.ac.in

**Ambha A**

IIT Bombay  
Mumbai, India, 400076  
ambha.career@gmail.com

**Ganesh Ramakrishnan**

IIT Bombay  
Mumbai, India, 400076  
ganesh@cse.iitb.ac.in

## Abstract

The wealth of information present in the World Wide Web has made internet search a de-facto medium for obtaining any required information. Users typically specify short and/or ambiguous queries and expect the answer to appear at the top. Hence, it can be extremely important to produce a diverse but relevant set of results in the precious top  $k$  positions. This calls for addressing two types of needs: (i) producing relevant results for queries that are often short and ambiguous and (ii) selecting a set of  $k$  diverse results to satisfy different classes of information needs. In this paper, we present a novel technique using a Biconvex optimization formulation as well as adaptations of existing techniques from other areas, for addressing these two problems simultaneously. We propose a graph based iterative method to choose diversified results. We evaluate these approaches on the QRU (Query Representation and Understanding) dataset used in SIGIR 2011 workshop as well as on the AMBIENT (Ambiguous Entities) dataset and present results on generating diversified query interpretations. We also compare these approaches against other online systems such as *Surf Canyon*, *Carrot2*, *Exalead* and *DBpedia* and empirically demonstrate that our system produces competitive results.

## 1 Introduction

The growth of internet has resulted in the proliferation of electronic documents on the World Wide Web. Every search engine, be it generic or application and domain specific, serves as a portal to access these documents. User queries, in general, are

short and often tend to be ambiguous and/or under-specified. In addition, a query can have multiple *concealed interpretations*. For example, *Sun* could be interpreted as “The sun as a star”, “Composition of Sun”, “Sun Micro systems company”, “Sun news paper”, “Sun Record music company”, and so on. We believe that, in addition to these concealed interpretations, *related interpretations* are also equally important. As examples, “Solar Cells” and “Photosynthesis”, could be interpretations related to this query. To improve user interaction and to guide him/her in further refining the query, it could help if the search engine generated these relevant interpretations as well. Due to the sheer size of online information and its diversity, the possible interpretations to a short query are enormous. In addition, users expect their intended answer to be present in the top few search results. This calls for presenting a diversified but relevant set of results in the top  $k$  positions. Note that, in this paper we consider the diverse search results produced by the search system as *interpretations* of the query in some sense. In addition, we consider each search result is a document describing some aspect related to the query. Hence we restrict our notion of interpretation to each such *document* in the search result.

We present an original method as well as adaptations of some existing methods to solve this problem. As for our proposed method, we construct an interpretation graph with potential interpretations as its nodes and edges indicating their similarity. Inspired by the works on GCD (Dubey et al., 2011) and MMR (Carbonell and Goldstein, 1998), we develop a new technique for diversity ranking of interpretations. As part of this technique, we propose an algorithm (Rel-Div) to learn the node and edge weights of the interpretation graph iteratively by solving a biconvex optimization (Gorski et al., 2007) problem. At query time, we solve a convex optimisation problem to choose

$k$  diverse nodes and present them as interpretations to the user query. We identify interpretations relevant to the query using a publicly available internet encyclopedia. Though we used Wikipedia as the source, we believe that the repository can be easily extended to accommodate other catalogs like YAGO and Freebase.

We compare our diversification approach with other diversification approaches (which were applied not necessarily to solve the same problem as ours) such as variants of GCD (Dubey et al., 2011), Affinity Propagation (Frey and Dueck, 2006),(Frey and Dueck, 2007). We evaluated results on benchmark queries from the SIGIR 2011 workshop’s QRU (Query Representation and Understanding) dataset and the AMBIENT data sets. In addition, we compare the diversity of interpretations generated by these approaches against those of other online systems such as Surf Canyon, Carrot2, Exalead and DBpedia (URLs of all these systems listed under References)

We summarize our contributions as: 1) Top-K diversity ranking using a graph based approach. 2) Iterative Graph weight learning technique - A new iterative technique for learning the node and edge weights for an interpretation graph by solving a biconvex optimisation problem.

The rest of the paper is organized as follows: In Section 2 we present related work. In Section 3 we describe our technique of iterative graph weight learning and diversity ranking. In Section 4 we demonstrate the utility of our technique by applying it to the interpretation generation task from Wikipedia. In Section 5, we present experimental evaluations. We conclude our work in the subsequent section.

## 2 Prior work

Most of the prior research has focused on generating diversified result urls. The approach presented by (Swaminathan et al., 2009) filters initial search results and covers diversified topics based on bag of words measures. Yisong and Joachims (Yue and Joachims, 2008) train a model using Struct SVM and encode diversity as a penalty function (this is penalty for not covering certain topics). Most recently, Brandt et al.(Brandt et al., 2011) and Raman *et. al.* (Raman et al., 2011) proposed an approach for *dynamic ranking* and then group URLs with similar intentions.(Dubey et al., 2011) formulate the problem of ensuring diversity as that of

identifying relevant urls which are most likely to be visited by the random surfer. We propose a new approach for interpretation generation. The report (Hearst, 2006) by M.A Hearst claims that clustering based on similarity measure may not always result in meaningful interpretations or labels. So, instead of dynamically generating labels, we pick labels or relevant interpretations for a query from the pool of labels. We use Wikipedia as a primary source to capture these interactions along with their semantic relations. (Hahn et al., 2010),(Ben-Yitzhak et al., 2008) produce Wikipedia pages as search results and align the search results along a set of fine grained attributes/facets. In our work, facets (which we refer to as interpretations) are neither predefined nor necessarily fine grained. Moreover, as we will see, our interpretations need not be restricted to Wikipedia entities. Closest to our approach is the approach of (Ma et al., 2010). They apply page ranking technique on the graph constructed using query log statistics to obtain diversified interactions.

## 3 Diversified Interpretation Generation

### 3.1 Our Problem

Given a large corpus  $U$  of documents and a short user query  $q$ , we define a function  $H(q, U)$  that returns a subset of documents  $S = \{e_1 \dots e_n\} \subseteq U$ , satisfying the query  $q$ . The function  $H(q, U)$  acts as a filtering function to retrieve the documents  $S$  that are syntactically and/or semantically related to the query  $q$ . In its simplest form,  $H(q, U)$  can just return  $U$  without performing any filtering, which is not generally useful. It is important to design an  $H(q, U)$  (*e.g.*, keyword based lookup, semantics matching, *etc.*) that can help reduce the search space in a meaningful manner. Our goal is to choose a set of  $k$  documents from  $S$  and we assume that to best satisfy the user intention, these  $k$  documents presented to the user should be diverse yet highly relevant to the query  $q$ .

### 3.2 The Training Algorithm

We expect groups of documents in  $S$  to be related to each other via some semantic relations. We initially construct a document-relation graph using  $e_1 \dots e_n$ . We refer to this graph as an *Interpretation Graph*, since the documents in this graph are obtained as various interpretations of the query. While the nodes are documents from  $S$ , each edge is a relation between the documents. A relation

could be one of synonymy, hyponymy, meronymy, homonymy, *etc.*. These relations could be obtained from external catalogs such as Wikipedia, Wordnet, *etc.*

Each node in the graph is assigned a score which represents the relevance of the node to the query. We use the notation  $b_q$  to represent the column vector (of size  $n \times 1$ ) containing all the node relevance scores. The weight on an edge represents the degree of similarity between the two nodes connected by that edge. We use the notation  $C_q$  (of size  $n \times n$ ) to represent the matrix of edge scores reflecting similarity between pairs of nodes. Note that, each column  $C_q^i$  of the matrix  $C_q$  represents a document  $e_i$  and the cell values in that column indicate the similarity of document  $e_i$  with other documents. The scores in  $b_q$  are used to ensure that the subset of  $k$  interpretations are relevant to  $q$ , whereas the similarity scores in  $C_q$  are used to ensure diversity in the subset of  $k$  interpretations.

We assume that we are provided training data, consisting of queries and their correct interpretations. Our goals in training are to 1) develop a model for the node score  $b_q$ , 2) develop a model for the edge potentials  $C_q$  and 3) learn parameters of these models such that the set of  $k$  relevant yet diverse nodes obtained from the graph using  $b_q$  and  $C_q$  are consistent with the training data. Thus, implicit in our third goal is the following subproblem, which is also our query time inference problem: 4) compute a subset of  $k$  best interpretations using  $b_q$  and  $C_q$ , that represent  $k$  diverse, but relevant interpretations. A part of the graph for the query "sun" is depicted in Figure 1

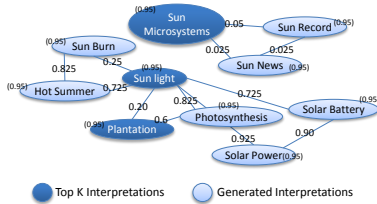


Figure 1: Interpretation Graph for the query *Sun*

### 3.2.1 Modeling node potentials ( $b_q$ )

In order to build a learning model for  $b_q$ , it is important to define a good set of features that characterize the node's relevance to the query. Let  $N_{1..|N|}(q, S)$  be a set of  $|N|$  query independent node features. Each feature  $N_f(q, S)$  evaluates the relevance of documents in  $S$  to the query  $q$  and returns a vector of scores. These feature functions are problem specific and crafted carefully to bring

out the relevance between query and documents (such as term overlaps, n-gram matches, etc). In Section 4 we provide some practical examples of node features.

The node potential vector  $b_q$  is obtained by combining the scores returned by individual feature functions  $N_f(q, S)$ . One of the obvious choices is to use Logistic Regression (Yan et al., 2003). *i.e.*  $b_q[i] = \frac{1}{1+e^{-\sum_{f=1}^{|N|} w_f N_f(q, S)[i]}}$ . The weight vector  $W^T = [w_1 \dots w_{|N|}]$  is learnt through supervised training explained in Section 3.2.3.

### 3.2.2 Modeling edge potentials ( $C_q$ )

To learn the edge potentials, it is important to define a good set of features that measure the similarities between every pair of nodes and return similarity scores. Higher the score, more similar are the nodes. Let  $C_{1..|C|}(S)$  be the set of  $|C|$  edge features that evaluate similarities between documents in  $S$  and each returns a  $n \times n$  matrix of scores. These feature functions are problem specific and crafted carefully to bring out the similarities between the documents. In Section 4 we provide some practical examples of edge feature construction using Wikipedia.

The edge potential matrix  $C_q$  is obtained as  $C_q = \sum_{f=1}^{|C|} \lambda_f C_f(S)$  where  $0 \leq \lambda_f \leq 1$  and  $\sum \lambda_f \geq 1 \forall f$ . The weight vector  $\lambda^T = [\lambda_1 \dots \lambda_{|C|}]$  is learnt through supervised training explained in Section 3.2.3.

### 3.2.3 Learning feature weights $W^T, \lambda^T$

Proposition 1:

$$b_q \approx \sum_{j=1}^k \tilde{C}_q^{i_j} \quad (1)$$

for sufficiently large  $k$  diverse documents, where,  $\tilde{C}_q$  is the matrix  $C_q$  with the columns scaled so that the diagonal cell values match the relevance value, *i.e.*,  $\tilde{C}_q(i, i) = b_q(i)$ . The values  $i_1 \dots i_k$  represent indices of  $k$  columns of matrix  $\tilde{C}_q$ . Hence,  $\tilde{C}_q^{i_j}$  is the  $i_j$ th column of matrix  $\tilde{C}_q$ .

The intuition behind this approximated equality comes from the fact that, two similar documents should have similar relevance score with the query and we are interested in selecting  $k$  diverse documents. Let  $e_i$  be one of these  $k$  diverse documents. If the documents  $e_{j_1} \dots e_{j_p}$  are similar to  $e_i$ , then,  $b_q[i] \approx b_q[j_1] \approx \dots \approx b_q[j_p]$  and  $C_q[i, i] \approx C_q[i, j_1] \approx \dots \approx C_q[i, j_p] \approx 1$  and  $C_q[t] \approx 0, t \notin j_1 \dots j_p$ . But, we already know that  $\tilde{C}_q[i, i] = b_q[i]$ . That implies,  $b_q[j_1] \approx \tilde{C}_q[i, j_1]$ ,

$b_q[j_2] \approx \tilde{C}_q[i, j_2], \dots, b_q[j_p] \approx \tilde{C}_q[i, j_p]$ . When we take the summation on all diverse  $k$  documents, the Equation 1 holds.

Based on the above proposition, we present an algorithm to learn weights  $W^T$  and  $\lambda^T$  iteratively in a supervised learning setup. The training data is provided in a vector  $r_q$  (of size  $n \times 1$ ) such that  $r_q[i] = 1$  if the document  $e_i$  is relevant to the query (and one of diverse documents), otherwise,  $r_q[i] = 0$ . Note that, the quantity  $\tilde{C}_q r_q$  represents the sum of  $k$  columns (assuming  $k$  number of 1s in  $r_q$ ) and is the RHS of Equation 1.

Our training objective is to learn  $\lambda^T$  and  $W^T$  such that Equation 1 holds. Formally, the problem being solved is:

$$\underset{\lambda_1, \dots, \lambda_{|C|}, w_1, \dots, w_{|N|}}{\operatorname{argmin}} D \left( \frac{1}{1 + e^{-\sum_g w_g N_g}}, \sum_f \lambda_f \tilde{C}_f r_q \right) \quad (2)$$

where  $D(x, y)$  is a distance measure between  $x$  and  $y$ . (for e.g., KL Divergence, Euclidean, etc.);  $\tilde{C}_f$  is the normalized  $C_f$  as in Proposition 1.

Applying the coordinate descent technique, we learn the weights  $W^T$  and  $\lambda^T$  iteratively using two steps outlined in Equation 3 and Equation 4, each of them convex in the respective optimization variables, hence our optimisation problem is biconvex.

div-step: Learn  $\lambda_1^{(t)}, \lambda_2^{(t)}, \dots$  holding  $w_1^{(t-1)}, w_2^{(t-1)}, \dots$  constant, by solving:

$$\underset{\lambda_1, \lambda_2, \dots}{\operatorname{argmin}} D \left( \frac{1}{1 + e^{-\sum_g w_g^{(t-1)} N_g}}, \sum_f \lambda_f^{(t)} \tilde{C}_f r_q \right) \quad (3)$$

rel-step: Learn  $w_1^{(t)}, w_2^{(t)}, \dots$  holding  $\lambda_1^{(t-1)}, \lambda_2^{(t-1)}, \dots$  constant, by solving:

$$\underset{w_1, w_2, \dots}{\operatorname{argmin}} D \left( \frac{1}{1 + e^{-\sum_g w_g^{(t)} N_g}}, \sum_f \lambda_f^{(t-1)} \tilde{C}_f r_q \right) \quad (4)$$

In *div-step*, we learn  $\lambda^T$  by holding  $W^T$  fixed and honoring Equation 1. In *rel-step*, we learn  $W^T$  by holding  $\lambda^T$  fixed. The relevance and divergence is enforced during training through the vector  $r_q$ .

We learn node and edge feature weights iteratively by recognizing and assigning weights to prominent node and edge features that satisfy queries of different types. Having all statistically driven computation of weights for edge features can minimize the side effect of poor node features and likewise computing weights for node features can decrease the consequences of poor edge features.

Algorithm 1 outlines the training procedure.  $I_q^+, I_q^-$  are the set of relevant and irrelevant documents for each query  $q$  in the ground truth that is used for training.

### 3.3 Query-time Inference

For a new user query  $q$ , inference problem is to choose  $k$  diversified results. Using  $H(q, \mathcal{C})$  we reduce the search space drastically and get the set  $\mathcal{S}$ . Otherwise, we need to run our inference on entire set  $U$ , which is very expensive. We then compute the node and edge feature matrices for all defined node and edge features. These individual feature matrices are then combined (using  $\lambda^T$  and  $W^T$ ) to obtain vector  $b_q$  and matrix  $C_q$ . Based on Proposition 1, our inference objective is to choose  $k$  columns from the matrix  $\tilde{C}_q$  such that their sum is as close as possible to  $b_q$ . Formally, the problem being solved is:

$$\underset{i_1, \dots, i_k}{\operatorname{argmin}} D \left( b_q, \sum_{j=1}^k \tilde{C}_q^{i_j} \right) \quad (5)$$

where  $i_1, \dots, i_k$  are indices of  $k$  columns of  $\tilde{C}_q$ .

Determining the exact solution (i.e.  $i_1, \dots, i_k$  columns) to the above optimization problem turns out to be computationally infeasible. Hence, we have to resort to an approximate solution. Algorithm 2 describes a greedy inference procedure. At each step we pick one column from  $\tilde{C}_q$  that minimizes the distance in Equation 5 most. However, we also ensure that the picked column is most diverse from the already selected columns in the previous steps. At the end of  $k$  steps we will have  $k$  diverse, but relevant documents.

#### Algorithm 1 Training

- 1: **Input:** Set of training data instances  $\{q, I_q^+, I_q^-, N_f, C_f, r_q\}$
- 2: **Output:**  $W^T$  and  $\lambda^T$
- 3: initialize variables  $W^T$  and  $\lambda^T$
- 4: learn initial  $W^T$  using Logistic Regression  $\triangleright$  uses  $\{q, I_q^+, I_q^-, N_f\}$   
 $\triangleright \tilde{C}_q, \tilde{C}_f$  used below are normalized  $C_q, C_f$  as in Proposition 1
- 5: **while** not converged( $|b_q - \tilde{C}_q r_q|$ ) **do**
- 6:  $b_q =$  compute relevance matrix using  $W^T$  and  $I_q^+$
- 7: find  $\lambda^T$  so that  $D(b_q, \sum_f \lambda_f \tilde{C}_f r_q)$  is minimized  $\triangleright W^T$  is fixed
- 8:  $p_q = \sum_f \lambda_f \tilde{C}_f r_q$
- 9: find  $W^T$  so that  $D\left(\frac{1}{1 + e^{-\sum_f w_f N_f}}, p_q\right)$  is minimized  $\triangleright \lambda^T$  is fixed
- 10: **end while**  
**return**  $(W^T, \lambda^T)$

#### Semantic relations and values from Wikipedia page excerpts

1. **Synonym:** All redirected names of the Wikipedia page.
2. **Association:** All valid hyperlinks of a Wikipedia page.
3. **Frequent:** All phrases occurring more than two times within a Wikipedia page section.
4. **Synopsis:** All nouns, verbs, adjectives from the abstract and titles of the sections in a Wikipedia page
5. **Hyponym:** All pages/sub categories of selected categories ending with Wikipedia page title. Ex: For Sony: robotics at Sony.
6. **Meronym:** All phrases which occur both in wordnet meronyms and with in Wikipedia pages.
7. **Hypernym:** All parent categories of selected categories.
8. **Homonym:** Pages referring to one or more disambiguation page.
9. **Sibling:** Siblings are the sub categories/pages which do not follow hyponym pattern. Ex: For Sony: list of sony trademarks

Table 1: Semantic Relations

#### Algorithm 2 Inference

- 1: **Input:** User query  $q$ , Corpus  $U$ ,  $\lambda^T$ ,  $W^T$ ,  $N_f, C_f$
- 2: **Output:**  $k$  diverse interpretations
- 3: Generate  $S = H(q, C)$  and build a graph using documents in  $S = \{e_1, \dots, e_n\}$
- 4: Compute  $b_q$  using  $W^T$  and node features  $N_{1..|N|}(q, S)$
- 5: Compute  $C_q = \sum_f \lambda_f C_f(S)$  and normalize as in Proposition 1
- 6:  $R = \{i_i \in Q\}$  ▷ set of selected indices
- 7:  $Q = \{i_1, \dots, i_n\}$  ▷ indices to select
- 8: **for**  $i = 1$  to  $k$  **do**
- 9: 
$$\underset{c_k \in Q/R}{\operatorname{argmin}} \left\{ D \left( b_q, \sum_{r \in R \cup \{c_k\}} (C_q^r) \right) \times \left( 1 - \frac{1}{2} \min \left( D(C_q^{R1}, C_q^{c_k}), \dots, D(C_q^{R|R|}, C_q^{c_k}) \right) \right) \right\}$$
▷ (query match)  $\times$  (dissimilar to selected),  $z$  is normalizer
- 10:  $R = R \cup \{c_k\}$
- 11: **end for**  
**return**  $k$  interpretations representing  $k$  columns  $R_1, \dots, R_{|R|}$

## 4 An example using Wikipedia

In this section we apply our Rel-Div technique to generate diverse but relevant results to a short and/or ambiguous user query using Wikipedia. For e.g. *Beagle*, *Laptop Charger*, *Sony Camera*, etc. We do not support queries which are highly rich in semantics like *Who invented music*, *Earn money at home* or very specific in nature like *DB2 error code 1064*.

In this case,  $U$  is a set of all Wikipedia entities (a.k.a. pages/articles). Note, in the context of Wikipedia, every document is treated as an entity. We defined  $H(q, U)$  as a set of filters which return Wikipedia entities  $S$ , called candidate interpretations, relevant to the user query. In order to build this filter function, we made use of prominent Wikipedia attributes (Title, Infobox entries, Frequently occurring words, etc) and different semantic relations between Wikipedia entities (Association via hyperlinks, Page Redirects, See Also links, etc). Table 1 summarizes these Wikipedia signals, which are captured for every entity.

## 4.1 Node Features

**Query Match:** Calculates the term overlap between query terms and the semantic relation terms of an interpretation. For e.g., for the query *Sony*, *PlayStation 2* is one of the interpretations, which has multiple occurrence of term *Sony* in one or more semantic relations.

**No. of Semantic Relation match:** Total number of semantic relations that contain the query terms. For e.g., for the query *Sony*, *PlayStation 2* interpretation may have 3 semantic relations (Synonym, Association and Frequent) containing term *Sony*.

**Title score:** Captures the interpretation title match to the query terms.

## 4.2 Edge Features

**Interpretation Content Overlap:** This feature measures the similarity between two interpretations by considering the amount of overlap between the words in these interpretation title and content.

**Decaying Recursive Similarity:** We considered neighborhood of an interpretation (hyperlinked entities, parent categories, subcategories, and grand parent pages) in the similarity measurement. However, an appropriate weight which decays with distance is set to avoid influence of farther neighborhood nodes.

**Link based proximity:** Determined by the depth  $D(lca)$  of the least common ancestor (LCA) of interpretations  $I_i$  and  $I_j$  from the root of Wikipedia category structure and the hop distance  $len(I_i, I_j)$  from  $I_i$  to  $I_j$  through LCA. Link proximity is defined as  $LP(I_i, I_j) \propto D(lca) * len(I_i, I_j)$ . When multiple LCAs exist, we define the proximity as  $\max(LP(I_i, I_j))$ .

## 5 Experimental Evaluation

We used Wikipedia as our knowledge source. We captured different signals shown in Table 1 for every Wikipedia entity.

### 5.1 Dataset

The QRU dataset used in SIGIR 2011 contains 100 TREC queries with various interpretations. We restricted our space of interpretations to Wikipedia entities. We also experimented with ambiguous queries from the AMBIENT dataset which contains 40 one word queries.

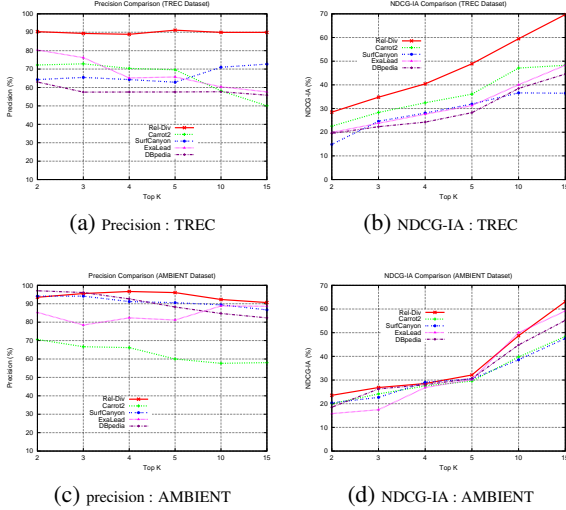


Figure 2: Comparison with external systems

		Precision(%)			Recall(%)			NDCG-IA(%)		
		@5	@10	@10	@5	@10	@10	@5	@10	@10
TREC	Rel-Div	<b>91.13</b>	<b>89.93</b>	<b>89.83</b>	<b>7.02</b>	<b>13.85</b>	<b>20.4</b>	48.9	59.47	<b>69.7</b>
	M-Div	89.87	84.27	84.32	6.74	12.71	18.88	<b>49.71</b>	<b>62.68</b>	67.39
	M-Div-NI	83.75	80	80	6.83	12.81	19.35	42.48	60.88	66.52
	AFP	78.3	76.9	80.7	6.3	12.4	18.1	34.2	38.8	47.6
AMBIENT	Rel-Div	96.05	92.3	90.67	7.33	<b>14.57</b>	<b>21.61</b>	<b>32.12</b>	<b>48.72</b>	<b>63.1</b>
	M-Div	96.15	<b>94.15</b>	<b>93.56</b>	<b>7.43</b>	14.37	21.61	32.41	47.49	58.09
	M-Div-NI	<b>96.2</b>	93.58	93.19	7.33	13.87	21.11	22.93	43.59	55.84
	AFP	88.4	90.9	92.3	6.9	13.6	21.47	32.09	45.9	55.1

Table 2: Results of different approaches

## 5.2 Evaluation methodology

Manually, interpretations for each query are marked as relevant or irrelevant and each interpretation is assigned one or more topics. The system is trained on 30 and tested on the rest. We evaluated results on queries of length one or two. The relevance of any interpretation to the query is measured using precision at different positions and the diversity is estimated using NDCG-IA (Agrawal et al., 2009). Recall measurement is tricky. It is practically not possible to manually inspect all Wikipedia entities and determine how many are actually relevant for a query. Hence we based our recall on the candidate interpretations generated. We manually counted number of relevant interpretations present in the candidate interpretations and measured how many of these relevant interpretations appeared in the top  $k$  interpretations.

In our experiments, we also consider a couple of other approaches to diversification, which have been reported in literature, though used in other problem settings. These include variants of GCD and affinity propagation (Frey and Dueck, 2006; Frey and Dueck, 2007).

**M-Div** : Uses page rank matrix  $M$  as in GCD instead of the  $C_q$  matrix.

**M-Div-NI** : Similar to M-Div, but node and edge weights are learnt independently, without any iter-

ations. This acts as GCD implementation.

**AFP:Exemplar** nodes of Affinity propagation are taken as interpretations.

## 5.3 Comparison with other approaches

While experimenting with our proposed approach, we found best performance when  $D$  in div-step was chosen to be KL-divergence and  $D$  in rel-step was chosen as the Euclidean distance. In Table 2, we compare the proposed diversification algorithm against M-Div, M-Div-NI and AFP on precision, recall and NDCG-IA measures.

We observed that our Ranking algorithm Rel-Div performs at par with (and sometimes even better than) M-Div and M-Div-NI. However, one of the major advantage of our method compared to M-Div and M-Div-NI is that, we need not calculate the inverse of  $C_q$  matrix, which is a computationally intensive process for a large dimension matrices. We conclude from the results that the Rel-Div performs consistently better than other approaches when both relevance and diversification are considered across all types of queries.

## 5.4 Comparison against other systems

We compare the diversity in search result using our approach against those from four other systems, viz., carrot2, SurfCanyon, Exalead and DBPedia to demonstrate that the Rel-Div approach produces high diversity in the search results, which is evident from the Figure 2.

## 6 Conclusion

We presented a body of techniques for generating top  $k$  interpretations to a user query using some internet encyclopedia, (in particular, Wikipedia was used in the experiments that were reported). Our approach is hinged on catering to two needs of the user, viz., that all the interpretations are relevant and that they are as diverse as possible. We addressed this using a bunch of node features and edge features based on semantic relations and learn these feature weights together iteratively. We present experimental evaluations and find that our approach performs well on both the fronts (diversity and relevance) in comparison to existing techniques and publicly accessible systems. We believe technique can be improved for better handling of multiword queries by adopting deep NLP parsing techniques, which will form part of our future work.

## References

- Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Jeong. 2009. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, New York, NY, USA. ACM.
- Ori Ben-Yitzhak, Nadav Golbandi, Nadav Har'El, Ronny Lempel, Andreas Neumann, Shila Ofek-Koifman, Dafna Sheinwald, Eugene Shekita, Benjamin Sznajder, and Sivan Yogev. 2008. Beyond basic faceted search. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 33–44, New York, NY, USA. ACM.
- Christina Brandt, Thorsten Joachims, Yisong Yue, and Jacob Bank. 2011. Dynamic ranked retrieval. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 247–256, New York, NY, USA. ACM.
- Surf Canyon. <http://www.surfcanyon.com/>.
- Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA. ACM.
- Carrot2. <http://search.carrot2.org/stable/search>.
- DBPedia. <http://dbpedia.org/facetedsearch>.
- Avinava Dubey, Soumen Chakrabarti, and Chiranjib Bhattacharyya. 2011. Diversity in ranking via resistive graph centers. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 78–86, New York, NY, USA. ACM.
- Exalead. <http://www.exalead.com/search/>.
- Freebase. <http://www.freebase.com/>.
- Brendan Frey and Delbert Dueck. 2006. Mixture modeling by affinity propagation. In *Advances in Neural Information Processing Systems 18*, pages 379–386. MIT Press, Cambridge, MA.
- Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315:2007.
- Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. 2007. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Math. Meth. of OR*, 66(3):373–407.
- Rasmus Hahn, Christian Bizer, Christopher Sahnwaldt, Christian Herta, Scott Robinson, Michaela BÄckerle, Holger DÄewiger, and Ulrich Scheel. 2010. Faceted wikipedia search. In Witold Abramowicz and Robert Tolksdorf, editors, *Business Information Systems*, volume 47 of *Lecture Notes in Business Information Processing*, pages 1–11. Springer Berlin Heidelberg.
- Marti A. Hearst. 2006. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49(4):59–61, April.
- Hao Ma, Michael R. Lyu, and Irwin King. 2010. Diversifying query suggestion results. In *AAAI*.
- Karthik Raman, Thorsten Joachims, and Pannaga Shivashwamy. 2011. Structured learning of two-level dynamic rankings. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 291–296, New York, NY, USA. ACM.
- Ashwin Swaminathan, Cherian V. Mathew, and Darko Kirovski. 2009. Essential pages. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 173–182, Washington, DC, USA. IEEE Computer Society.
- YAGO. <http://www.mpi-inf.mpg.de/yago-naga/>.
- Lian Yan, Robert H. Dodier, Michael Mozer, and Richard H. Wolniewicz. 2003. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *ICML*, pages 848–855.
- Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural svms. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 1224–1231, New York, NY, USA. ACM.



# Learning Based Approaches for Vietnamese Question Classification Using Keywords Extraction from the Web

Dang Hai Tran<sup>1</sup>, Cuong Xuan Chu<sup>1</sup>, Son Bao Pham<sup>1</sup>, Minh Le Nguyen<sup>2</sup>

<sup>1</sup>University of Engineering and Technology, Vietnam National University

<sup>2</sup>Japan Advanced Institute of Science and Technology

<sup>1</sup>{dangth, cuongcx\_54, sonpb}@vnu.edu.vn

<sup>2</sup>nguyenml@jaist.ac.jp

## Abstract

This paper presents our research on automatic question classification for Vietnamese using machine learning approaches. We have experimented with several machine learning algorithms utilizing two kinds of feature groups: bag-of-words and keywords. Our research focuses on two most important tasks which are corpus building and features extraction by crawling data from the Web to build a keyword corpus. The performance of our approach is promising where our system's precision outperforms the state-of-the-art Tree Kernel approach (Collins and Duffy, 2001) on a Vietnamese question corpus.

## Keywords

keyword collection, machine learning, Vietnamese question classification corpus.

## 1 Introduction

Question Classification (QC) is a task that, given a question, maps it to one of the predefined  $k$  classes, which indicates a semantic constraint on the sought-after answer (Li and Roth, 2006).

In a question answering system, before finding an answer of a question, the system has to identify which category it is asking about, and this is the obligation of question classification. Then, based on the identified category, we can narrow the space of answers that we have to find. Let us consider some examples:

- **Q:** *"Ai là người phụ nữ đầu tiên hy sinh trong chiến tranh Việt Nam?"* ("Who was the first woman killed in the Vietnam War?"), we expect to know that the target of this question is a **person**, thus reducing the number of possible answers significantly.

- **Q:** *Tại sao nắng màu vàng?* (*Why is the sunshine yellow?*) indicates that this question wants to get information about **reason**, thus in next steps our system just concerns about reason answers space rather than human or number categories.

The problem is that if the number of categories is more and the categories are more specific, the question answering system will spend more time classifying questions but its performance will be better. Let consider another example, in the next two questions, they are both asked to get information about **location**:

- **Q:** *Thành phố nào ở Canada có nhiều dân nhất?* (*What Canadian city has the largest population?*)
- **Q:** *Đất nước nào trao tặng Mỹ tượng nữ thần tự do?* (*Which country gave New York the Statue of Liberty?*)

More particularly, we can see that the target of the first question is a city and the other one is a country, both city and country are locations. In this case, location is considered a coarse-grained class and city and country are fine-grained classes. Naturally, the more fine-grained classes we have, the more difficult it is to tell them apart. For hierarchical categories, we adopted a two-level learning approach: first we solve the problem of classifying questions in the coarse-grained classes then based on this predicted label we continue with the fine-grained categories.

Our main contributions are building a Vietnamese corpus, collecting and using a kind of important features, which is keyword, for Vietnamese question classification. We tackle the corpus building by translating an existing well known English question corpus to Vietnamese. To the best of our knowledge, there is no publicly available Vietnamese questions corpus. As a result of

this work, we will share our newly built corpus to the research community. Collecting and using keywords is another important contribution of our work, which indicates that keyword extracted from Web is the most effective feature for Vietnamese question classification task. In this paper, we are going to present how we collect and extract keyword features for Vietnamese question classification.

The paper is organized as follows: Section 2 describes the related works and section 3 presents the process we take to build the Vietnamese data corpus. In section 4 we describe our Vietnamese question classification system, especially the features extraction step which involves crawling data from the Internet to create keyword features. In section 5 we show our experiments when using different machine learning algorithms with the set of features we extracted on our data corpus to classify questions in Vietnamese. Finally, section 6 provides some conclusions.

## 2 Related Works

There are many different approaches to resolve the question classification problem. Zhang and Lee (Zhang and Lee, 2003) with SVM (Cortes and Vapnik, 1995) using Tree Kernel, Li and Roth (Li and Roth, 2006) with SNoW model are two state-of-the-art approaches for English question classification. In Zhang and Lee's approach, the input question for this method is parsed into a syntax tree and converted to a vector in a multi-dimensional space. They introduced a new kernel function for SVM, Tree Kernel, constructed by dynamic programming to derive the similarity between two different syntax trees. This method achieved a precision of 90% with coarse-grained classification on TREC but there isn't any published results with fine-grained classification. In Li and Roth's approach, a set of features they used not only include syntactic features such as chunking but also include semantic features by using WordNet for English and building class-specific related words. Using semantic features, this method achieved a high precision of 92.5% with coarse-grained class and 85% with fine-grained class classification on a set of data including 21500 training questions from TREC 8, 9 (Voorhees, 1999; Voorhees, 2000), USC (Hovy et al., 2001), and 1000 testing questions from TREC 10, 11 (Voorhees, 2001; Voorhees, 2002).

## 2.1 Question Classification in Vietnamese

Question Classification in English is a classical problem but in Vietnamese, it is still a relatively new problem. To the best of our knowledge, there is not any research which works on open-domain Vietnamese question classification using learning based approach is published. In a research about question answering system for Vietnamese (Tran et al., 2009), the authors used machine learning approach for question classification module, however, the questions are only on travelling domain. Moreover, there is another research also working on question answering in Vietnamese (Dat et al., 2009). This system also focuses on answering questions on a specific domain and the authors used rule-based approach to resolve question classification module.

Combining the strengths of many solutions applied for English and the idea of (Tran et al., 2009), we started our research and experiment in Vietnamese question classification using machine learning approaches. Our main contributions are building a question set and a class-specific related word (keywords) set for Vietnamese.

## 3 Corpus Building

### 3.1 Question Hierarchy

From 1999, to support competitive research on question answering, The Text Retrieval Conference (TREC) has launched a QA track (TREC 8). Because the TREC QA track builds a fully automatic open-domain question answering system, there are many researches using TREC as experimental data sets. Importantly, the question type taxonomies of TREC can be used for any language. Besides there is not any standard for Vietnamese question classification yet, so we decide to use the question type taxonomies of TREC in our research.

TREC defined a two-layered taxonomy, which represents a natural semantic classification for typical answers. The hierarchy contains 6 coarse-grained classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine-grained classes. Table 1 shows the hierarchy of these classes in nearly 5500 training and 500 testing questions of TREC 10. Each coarse-grained class contains a non-overlapping set of fine-grained classes.

Coarse	Fine-Grained
ABBR	abbreviation, expansion
DESC	definition, description, manner, reason
ENTY	animal, body, color, creation, currency, disease/medical, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
HUM	description, group, individual, title
LOC	city, country, mountain, other, state
NUM	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight

**Table 1: The TREC coarse and fine-grained question classes.**

### 3.2 Question Translation

From the English question corpus, we translated them into Vietnamese and use them for Vietnamese question classification based on following rules:

- The content of Vietnamese question must correspond to its label.
- Some named entities can be changed without keeping the semantic meaning. For example, Washington can be changed to be Hà Nội or White House can be changed to Hồ Gươm.
- Given an English question, we can translate it into many Vietnamese questions with the same meaning but different in syntactic structures. As a result, our classifier can detect many kinds of Vietnamese questions.

In this project, we allocated 5 students who have TOEFL score > 500 for translating 6000 TREC English questions into Vietnamese in about 2 months. Every member of our group not only has to translate but also review the translated Vietnamese questions from other members to find mistakes to correct them and assure the quality of the translated data.

With our Vietnamese question classification problem, there is no publicly available Vietnamese corpus, when our corpus is made publicly available, it can be used for many works in the future.

## 4 Vietnamese Question Classification System (VnQCS)

### 4.1 Vietnamese Question Classification System Architecture (VnQCS - Architecture)

There are two main components in our system, the Feature Extractor and the Classifier (Figure 1). Source code of our system and the data corpus were made publicly available at: <https://code.google.com/p/vn-qcs/>

The Classifier contains two levels, the coarse-grained classification and the fine-grained one. In the first step, questions are classified into the coarse-grained classes, then taking result of the first step as a feature, we continue classifying questions to the fine-grained classes.

With a classification problem using machine learning approaches, the feature extraction is a key step. The quality of the set of features extracted directly impacts the classification precision. The Feature Extractor module consists of two components: Vietnamese Word Segmenter and Keyword Collector. Among them, Keyword Collector plays an important role in feature extraction method and it is the highlight of this research.

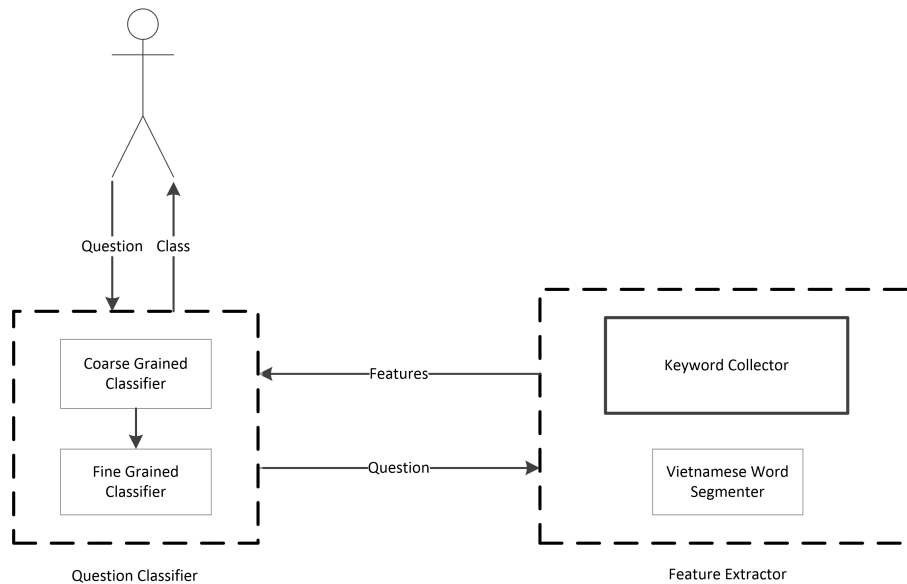
### 4.2 Feature Extraction in VnQCS

First, it is known that the linguistic characteristics of Vietnamese is different from English. Unlike English, in Vietnamese, one word may contain more than one token. For example, *mobile* (English) is translated into *điện thoại* (Vietnamese) and *mobile* is a word but *điện thoại* is a word which includes two tokens (*điện, thoại*). So, to match the characteristics of Vietnamese grammar, we will use words as a feature of algorithm in question classification.

We divide the features we extracted into three groups: bag-of-words, keywords and syntactic trees.

With **bag-of-words**, to get a set of features of Vietnamese words, we use a VnWordSegmenter tool<sup>1</sup> to extract them from Vietnamese training questions. With VnWordSegmenter tool, a question, for instance, "Bốn hình thức tồn tại của vàng là gì? (What four forms does gold occur in ?)", will be segmented into a sequence of words, as "Bốn hình\_thức tồn\_tại của vàng là\_gì?".

<sup>1</sup>Developed tool of iTim Company, website: <http://coccoc.com/about/>



**Figure 1: Architecture of VnQCS.**

With **keywords**, there are many keywords for a question class. In Li and Roth (Li and Roth, 2006), they used WordNet for English and built a set of class-specific words as semantic features and achieved the high precision (see section 2). But for Vietnamese, there isn't any lexical database like WordNet, so we have to develop an algorithm which collect keywords from the Internet which are lists of class-specific words for a smaller scale and useful enough for question classification (see Section 4.2.1). Moreover, we also manually collect keywords by observing the set of training questions. For examples, in reason class, with a why-question, we usually use "tại sao (why)" to start the question, or in abbreviation class, we usually use "viết tắt (abbreviation)". Most of classes have some specific words. This really has a significant impact when we use these lists of words as semantic features in question classification using machine learning.

With **syntactic trees**, the reason we extract this kind of features is that in Zhang and Lee (Zhang and Lee, 2003), the SVM algorithm using Tree Kernel method is the state of art for English question classification. So, we intend to answer the following question: "Does this method still outperforms the other methods in Vietnamese?". To use Tree Kernel, the questions must have corresponding syntax tree forms. For this, we used a parser tool for Vietnamese, Coltech-Parser (Le et al., 2009). The Coltech-Parser requires word segmentation for each input sentence. So, before using

the parser, we used the VnWordSegmenter tool to segment input questions.

#### 4.2.1 Keyword Collection from Web in VnQCS

Keywords are important semantic features though collecting them is not an easy task. Since the Internet contains a great number of web pages, this huge resource will help us to address the above-mentioned problem. The algorithm 1 describes how we collect keywords from Web and we hope it will be the basis for building semantic lexicons for other language processing tasks in Vietnamese.

- Firstly, we manually collect websites from the Internet which focus their content to one of our classes. Note that these websites should totally contain web content about one kind of class we want. As a result, our keywords from these will correspond to that class only and good for training features.
- In the next step, we crawl all links in these websites, note that we use only internal links since some external links will lead to other websites which are not related to the class we are interested in. Besides, in some cases all links that we want are only part of a website, we have to detect the format of them to get good content for training data.
- Base on links from previous step, we segment their text content. However, the words that we get in this step may contains some

**Input:** Set of websites which their contents focus on specific fine-grained class input

**Output:** Good keywords set for fine-grained class input

```
setOfPages = {};  
foreach website in websitesInputSet do  
    foreach internalLink in website do  
        | setOfPages = setOfPages + pageOfThisInternalLink;  
    end  
end  
  
setOfKeywords = {};  
foreach page in setOfPages do  
    remove all tags from page to get pageContent;  
    segment pageContent to get listOfWords;  
    foreach word in listOfWords do  
        | setOfKeywords = setOfKeywords + word;  
        | wordFrequency[word] = wordFrequency[word] + 1;  
    end  
end  
  
sort all words in decreasing order of wordFrequency and save result to sortedWords;  
eliminate all words which have low IDF index from sortedWords;  
choose top words from sortedWords and save result to keywordsResult;
```

**Algorithm 1:** Collect Keywords of a Specific fine-grained Class

trivial words like "là" (is), "và" (and), "hoặc" (or)... Since these stop words have low meaning, we can threshold their  $IDF^2$  index and eliminate small value ones. Besides, we count the frequency of each word in each class and choose top N words with biggest frequencies. Finally, reviewing these N words and removing unsuitable ones for target class are necessary works that we have to do.

## 5 Experiments

This section describes our experiments on the Vietnamese corpus that we built. These experiments will help us find out the most suitable combination of algorithms and features for this task on this dataset. The results of experiments indicate that the semantic features, especially keywords, are really useful.

We designed three experiments to test the precision of our classifiers on Vietnamese questions corpus which we built. The corpus consists of two data sets: a training data set which includes nearly 5500 questions and a testing data set which includes 500 questions translated from TREC 10 and all of questions are in 6 coarse-grained classes or 50 fine-grained classes. To evaluate the experi-

mental results, we use two main measures: weight average precision and weight average recall, which are all micro-average values.

- The first experiment evaluates the individual contribution of different feature types to question classification precision. In particular, we use Weka (Hall et al., 2009) to run machine learning algorithms namely Decision Tree (DT) (Quinlan, 1986), Naive Bayes (NB) (Bayes, 1763), SVM and Voting (Parhami, 1994), which are trained from our data we built using the following feature set: bag-of-words.
- In the second experiment, both bag-of-words and keyword features will be used together on machine learning algorithms on Weka. The goal is to verify the contribution of keyword features to question classification precision.
- Finally, we experiment with syntactic tree features set on SVM-Light-TK (Moschitti, 2004). This test will show us different affection of syntactic features between English and Vietnamese to question classification precision, and the important role of semantic features in Vietnamese question classification.

<sup>2</sup><http://en.wikipedia.org/wiki/Tf%E2%80%93idf>

## 5.1 Bag-of-words

	6 class		50 class	
	Precision	Recall	Precision	Recall
<b>Decision Tree</b>	87.3%	87%	78.2%	76.2%
<b>Naive Bayes</b>	84.4%	83.6%	78.2%	73%
<b>SVM</b>	<b>91.2%</b>	<b>91%</b>	<b>83.1%</b>	<b>82.4%</b>
<b>Majority Voting</b>	91%	90.6%	81.1%	79.4%

**Table 2: The question classification results using different machine learning algorithms, with same kind of feature: bag-of-words.**

Like results in (Zhang and Lee, 2003), table 2 shows us that in Vietnamese question classification, SVM still outperforms other methods with the same kind of features. With bag-of-words features, SVM model achieves the highest precision with 91.2% on coarse-grained class and 83.1% on fine-grained class classification.

## 5.2 Bag-of-words + Keywords

	6 class		50 class	
	Precision	Recall	Precision	Recall
<b>Decision Tree</b>	86.2%	86.2%	80.3%	77.4%
<b>Naive Bayes</b>	87.4%	86.2%	81.1%	78.4%
<b>SVM</b>	<b>94.1%</b>	<b>94%</b>	<b>85.4%</b>	<b>83.8%</b>
<b>Majority Voting</b>	94.1%	94%	83.5%	81.8%

**Table 3: The question classification results using different machine learning algorithms, with same kind of features: bag-of-words and keywords.**

If only using bag-of-words, we can not fully exploit the semantic elements of the language in Vietnamese. As we expected, with both bag-of-words and keyword features used together, although the precision of classification using DT or NB only increases slightly, it increases significantly if we use SVM. In particular, with 6 coarse-grained classes, the precision increases from 91.2% (table 2) to 94.1% (table 3), and with 50 fine-grained classes, it increases from 83.1% (table 2) to 85.4% using SVM (table 3). So, keyword features have an important role to increase the precision of question classification.

## 5.3 Tree Kernel

We experimented SVM-Light-TK to using SVM combined Tree Kernel for Vietnamese question classification since this state of the art method is successful for English data. However, SVM-Light-TK can only classify binary label, we use one-vs-all strategy for problems of 6 coarse-grained

	6 class		50 class	
	Precision	Recall	Precision	Recall
<b>Tree Kernel</b>	88.4%	88%	75.1%	67.4%
<b>Bag-of-words</b>	91.2%	91%	83.1%	82.4%
<b>Bag-of-words + Keywords</b>	<b>94.1%</b>	<b>94%</b>	<b>85.4%</b>	<b>83.8%</b>

**Table 4: The question classification results using SVM algorithm with some different kinds of features.**

classes and 50 fine-grained classes. The precision of this taxonomy is 88.4% for coarse-grained classes but only 75.1% for fine-grained classes (see table 4).

## 6 Conclusion

There are two main contributions of this paper. Firstly, we created a corpus for Vietnamese question classification (section 3). All the English questions in TREC 10 were translated into Vietnamese not only for this research but also many works in the future. This corpus will be made publicly available. Secondly, we extracted several feature groups and found out that the semantic features (bag-of-words and keywords) are really helpful to Vietnamese question classification meanwhile syntactic ones (syntactic tree) don't contribute so much to taxonomy precision. So we propose a method for collecting keywords from the Internet in a large scale. There isn't any WordNet for Vietnamese but with this method, we still have enough training data features for classifying a large range of Vietnamese questions.

Though Vietnamese question classification is a new challenge and there is not any work done on this, our experimental results indicate that the Vietnamese question classification can be addressed with relatively high precision using machine learning approaches. The result of classification can achieve a high precision of 94.1% with coarse-grained class classification and 85.4% with fine-grained class classification.

## Acknowledgments

We wish to thank The Vietnam National Foundation for Science and Technology Development (NAFOSTED) for financial support. Thanks also to NLP group at University of Engineering and Technology for partially supporting us in building the corpus. Finally, we thank the anonymous reviewers for helping us improve the presentation.

## References

- T. Bayes. 1763. An essay towards solving a problem in the doctrine of chances. In *Philosophical Transactions of the Royal Society*, volume 53, pages 370–418.
- M. Collins and N. Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS14)*.
- C. Cortes and V. N. Vapnik. 1995. Support vector machines. In *Machine Learning*, pages 273–297.
- Q. N. Dat, Q. N. Dai, and B. P. Son. 2009. A vietnamese question answering system. In *International Conference on Knowledge and Systems Engineering, KSE*, pages 26–32.
- M. Hall, E. Frank, G. Holmes, B. Pfahringer, R. Reutemann, and I. H. Witten. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations*, 11:10–18.
- E. Hovy, L. Gerber, U. Hermjakob, C. Lin, and D. Ravichandran. 2001. Toward semantics-based answer pinpointing. In *the DARPA HLT conference*.
- A. C. Le, P. T. Nguyen, Vuong H. T., M. T. Pham, and T. B. Ho. 2009. An experimental study on lexicalized statistical parsing for vietnamese. In *KSE '09 Proceedings of the 2009 International Conference on Knowledge and Systems Engineering*, pages 162–167, Washington, DC, USA.
- X. Li and D. Roth. 2006. Learning question classifiers: The role of semantic information. *Nat. Lang. Eng.*, 12(3):229–249.
- A. Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42-th Conference on Association for Computational Linguistic*, Barcelona, Spain.
- B. Parhami. 1994. Voting algorithms. In *IEEE Transactions on Reliability*, volume 43, pages 617–629.
- J. R. Quinlan. 1986. Induction of decision trees. In *Machine Learning 1*, pages 81–106. Kluwer Academic Publishers.
- M. V. Tran, D. V. Nguyen, T. O. Tran, T. U. Pham, and Q. T. Ha. 2009. An experimental study of vietnamese question answering system. In *IALP '09 Proceedings of the 2009 International Conference on Asian Language Processing*, pages 152–155, Washington, DC, USA.
- E. Voorhees. 1999. The trec-8 question answering track report. In *Proc. of 8th Text Retrieval Conference, NIST*, pages 77–82, Gaithersburg, MD.
- E. Voorhees. 2000. Overview of the trec-9 question answering track. In *Proc. of 9th Text Retrieval Conference, NIST*, pages 71–80, Gaithersburg, MD.
- E. Voorhees. 2001. Overview of the trec 2001 question answering track. In *Proc. of 10th Text Retrieval Conference, NIST*, pages 157–165, Gaithersburg, MD.
- E. Voorhees. 2002. Overview of the trec 2002 question answering track. In *Proc. of 11th Text Retrieval Conference, NIST*, pages 115–123.
- D. Zhang and W. S. Lee. 2003. Question classification using support vector machines. In *Proc. of SIGIR*, pages 26–32.

# Detecting Bot-Answerable Questions in Ubuntu Chat

**David C. Uthus**

NRC/NRL Postdoctoral Fellow  
Washington, DC 20375  
duthus@google.com

**David W. Aha**

Navy Center for Applied Research in AI  
Naval Research Laboratory (Code 5514)  
Washington, DC 20375  
david.aha@nrl.navy.mil

## Abstract

Ubuntu’s Internet Relay Chat technical support channel has bots that output specific messages in response to command words from other channel users. These messages can be used to answer frequently-asked questions instead of requiring an expert to (repeatedly) type a lengthy reply. We describe an approach to automatically distinguish bot-answerable questions, which would mitigate this problem. To the best of our knowledge, this is the first work on investigating question answering in a multiparticipant chat domain. Our results indicate that for some types of questions, supervised learning algorithms perform well on this task and, in addition, that character  $n$ -grams are a better representation than traditional bag-of-words for this task and domain.

## 1 Introduction

Ubuntu (a Linux-based operating system) maintains multiple Internet Relay Chat (IRC) channels for technical support. Some of these channels contain bots, which are automated agents pre-programmed to perform certain tasks. One of the bots can output pre-written messages, called *factoids*, in response to command words. For example, if a user types “!flash”, then the bot would output “To install Flash see <https://help.ubuntu....mats/Flash> - See also !Restricted and !Gnash”. These factoids are used to answer common questions, enforce channel guidelines, direct non-English speakers (in their native tongue) to Ubuntu’s foreign language support channels, and query Ubuntu’s repository of packages. While useful, this bot must be manually invoked. Automating the bot to self-detect and answer questions that it can answer could reduce

the workload for knowledgeable experts trying to help other users. This is applicable to not only Ubuntu’s technical support channels, but to other IRC channels providing technical support (e.g., Debian’s support channels) and to channels that use similar bots (e.g., Eggdrop and Infobot) for other purposes.

We describe initial steps on a self-invoking bot. We begin by investigating the multi-classification task of which questions are bot-answerable questions (BAQ) and which are human-answerable questions (HAQ), which requires a human to answer due to the question’s complexity. We implemented a baseline non-learning approach and supervised support vector machines (SVM) and  $k$ -nearest neighbor ( $k$ -NN) algorithms. Our results show that a bot can answer with confidence some types of questions, especially those directing users to more appropriate channels for help on certain topics.

Our contributions are as follows:

- **Problem:** We identified a real-world problem that has not been investigated, despite bots having been around for years on IRC channels.
- **Data:** We created an annotated multiparticipant chat corpus that is publicly available.
- **Empirical study:** We report on our investigation of applying supervised learning algorithms and leveraging different feature representations, whose results will be used as a foundation for a larger case-based reasoning approach.
- **Discussion:** We describe how some types of automatically-answerable questions can be easy or difficult to classify.



## 2 Related Work

Chat is a difficult medium to analyze: its characteristics make it difficult to apply traditional natural language processing techniques. It has uncommon features such as frequent use of abbreviations, acronyms, missing subject pronouns, emoticons, abbreviated nicknames, words stripped of vowels to reduce number of keystrokes, and entangled conversation threads (Uthus and Aha, 2013a).

In the multiparticipant chat domain, there has been some work in creating a bot that can answer questions with Cobot (Isbell et al., 2006). This bot was limited in capability – it could only respond to questions directed at it. Another recent work resulted in a bot which could respond to utterances through word matching and used templates for output (Shaikh et al., 2010).

Also related in this domain are a few military research efforts that have focused on classifying chat messages. One examined profile-driven information extraction from chat using regular expressions and entity classes (Berube et al., 2007). Another examined identifying uncertainty and urgency within a chat message using rule-based approaches and statistical analysis (Budlong et al., 2009). A third, whose work is most similar to ours, compared several supervised algorithms for classifying chat utterances (Dela Rosa and Ellen, 2009). Using an artificial chat log, they classified messages as either non-important filler messages or as messages containing Navy ship updates. Their results showed  $k$ -NN and SVM performed best for this task. Our task differs from these previous investigations in that we are applying supervised learning algorithms to a multi-labeled corpus composed of real chat messages.

This problem is also related to the larger field of question answering, such as pertaining to discussion boards (Hong and Davison, 2009; Kim et al., 2007), frequently asked question files (Burke et al., 1997), and community-based question answering (Zhou et al., 2012). An important difference between this body of related work and what we are investigating is the medium. Multiparticipant chat is more difficult to work with compared to other mediums due to entangled conversation threads – a researcher cannot easily automatically analyze the messages of a single conversation. In prior work, researchers could usually isolate individual conversations automatically, making it possible to identify (to some extent) the question and

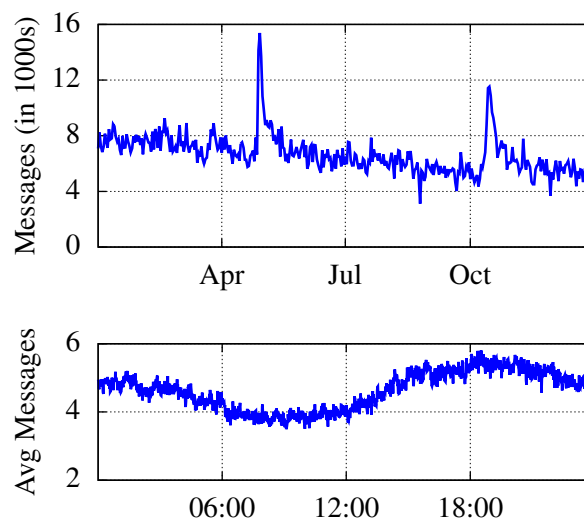


Figure 1: The volume of messages in the IRC channel #ubuntu during 2011.

the answer. Another important difference is the temporal scale – chat is in real-time, and a chat user expects to receive an answer quickly. A chat user can only see messages while they are logged in (in the case where there is no archive being stored offline). Both of these differences results in new challenges not seen in other mediums for question answering.

## 3 Ubuntu’s IRC Channels

The IRC channel #ubuntu is Ubuntu’s primary technical support channel. It provides support for those who have problems using Ubuntu; it is not used for socializing or for receiving help with other Linux distributions (e.g., Debian, Linux Mint, Fedora) or software.

The channel’s traffic level varies throughout the year and day (see Figure 1). During the year, it experiences heavy traffic during Ubuntu’s semi-annual new releases in April and October and generally experiences heavy traffic during the North American and European evening hours. Heavy traffic creates difficulties for users trying to get answers to their questions.

ubottu<sup>1</sup> is the bot that can access 1234 factoids, corresponding to 2324 commands (some factoids are mapped to multiple commands). It can also provide information about any software package found in Ubuntu’s software repository. A channel user (oftentimes an expert) can task ubottu to answer another user’s question (see

<sup>1</sup><http://ubottu.com/>

```

[13:19] <p5yx> is the netbook
remix not available anymore?
[13:20] <histo> !unr | p5yx
[13:20] <ubottu> p5yx: Starting
with Ubuntu 11.04, the Ubuntu
Netbook Edition is no longer
being offered as a separate
install as Unity is now standard
for all Ubuntu desktop installs.

```

Figure 2: Example of `ubottu` being invoked with a command word (in this case “!unr”) to answer a question.

Figure 2). Automating `ubottu` would allow experts to focus their valuable time on responding to more challenging requests.

## 4 Corpus

We created an annotated corpus by pulling questions from the Ubuntu Chat Corpus, specifically from the `#ubuntu` channel logs (Uthus and Aha, 2013b). This corpus has 4577 messages, including 2002 HAQs and 2575 BAQs from 68 factoid categories. These messages were taken from chat logs from 28 April 2011 (the day Ubuntu 11.04 was released) to 13 October 2011 (the day before Ubuntu 11.10 was released).

We looked for messages in which a question is answered with a factoid, or a question required a human to answer. To judge between these two types, we relied on the expertise of users and how they answered the questions. To reduce noise, we limited HAQs to conversations in which the first reply came from a user who invoked `ubottu` frequently (i.e., experts). These `ubottu` invokers are considered a better judge of what is a BAQ or HAQ compared to someone who rarely invokes `ubottu` to answer questions. For the BAQs, we restricted questions to those with at least ten examples mapped to a factoid. Figure 3 shows the distribution of BAQs to factoids in our corpus.

Some of the corpus’ messages are not in English. In such cases, users will be directed to one of Ubuntu’s foreign-language support channels (though a user can re-ask their question in English). Some languages present in the corpus include Spanish, French, Chinese, Russian, German, Polish, and Portuguese. Additionally, some of these messages are written with non-Latin characters, such as Chinese and Russian.

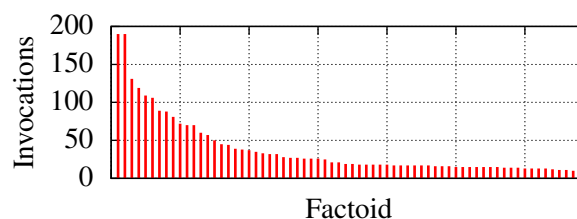


Figure 3: Distribution of the sixty-eight factoid invocations in our corpus.

## 5 Approach

We are using  $k$ -NN and SVM algorithms for classifying messages. This builds on results by Dela Rosa and Ellen (2009), who had found these two supervised learning algorithms to work best on chat messages. Implementation of these algorithms were obtained from Scikit-learn (Pedregosa et al., 2011).

For preprocessing, text was normalized by lowering the case for each term.

We examined different representations for encoding the questions. These include bag-of-words, bigrams, and character  $n$ -grams. With the character  $n$ -grams, we examined  $n$ -grams which overlap words and  $n$ -grams which are restricted to within word boundaries. We used  $tf - idf$  to weigh the features and  $\chi^2$  for feature selection.

## 6 Empirical Study

We have two hypothesis we are testing:

**H1:** Supervised learning algorithms will outperform a non-learning baseline approach for classifying BAQs.

**H2:** Using character  $n$ -grams for this domain will allow for better precision and recall when compared to more traditional representations of bag-of-words and bigrams.

Our intuition for H2 is that we believe that character  $n$ -grams will allow for better representation of misspelled words commonly seen in chat messages when compared to bag-of-words and bigrams.

### 6.1 Baseline

Our non-learning baseline algorithm maps questions to factoids by checking if the question contains the factoid command as a word token. As a reminder, multiple factoids can map to the same response. If a question contains multiple factoids, then the most frequently invoked factoid is applied (ties are broken by alphabetical ordering). If a

Representation	$\chi^2$ Feature Size	Precision	Recall	F <sub>0.5</sub> Score
Non-learning baseline				
–	–	0.44	0.24	0.37
SVM				
Character 3-grams, WB	4000	0.67	0.42	0.60
Character 3-grams	3200	0.67	0.38	0.59
Character 4-grams, WB	3600	0.63	0.40	0.57
Bag-of-words	1600	0.62	0.40	0.56
Character 4-grams	4000	0.58	0.35	0.51
Bigrams	1200	0.51	0.20	0.39
<i>k</i> -NN				
Character 4-grams, WB	800	0.55	0.34	0.49
Character 3-grams, WB	800	0.55	0.32	0.48
Character 4-grams	1200	0.57	0.30	0.48
Character 3-grams	800	0.54	0.41	0.47
Bag-of-words	400	0.54	0.31	0.47
Bigrams	400	0.44	0.15	0.32

Table 1: Results for the baseline, SVM and *k*-NN algorithms. WB means the character *n*-grams were bounded within word boundaries.

question contains no factoids, then it is considered a HAQ.

## 6.2 Metrics

We used a 10-fold cross evaluation protocol and precision, recall, and the F<sub>0.5</sub> score as our evaluation metrics. For this work, we consider precision to be more important than recall because a bot that frequently answers questions incorrectly could anger chat users and cause them to ignore the bots. Therefore, F<sub>0.5</sub> is more appropriate here than the standard F score because it places more emphasis on precision. Additionally, as these are multi-classification problems, we used the macro version of these metrics to average over the different labels.

When calculating precision, recall, and F<sub>0.5</sub>, we omit the HAQ scores when calculating the macro scores for this multi-classification problem. We also omitted any questions that are incorrectly labeled as HAQ for calculating precision and recall scores because a HAQ can be answered by a human expert. Essentially, we do not penalize for erring on the side of caution.

## 6.3 Results

Table 1 summarizes the results of the application of our baseline and learning algorithms. We ap-

plied all variations of the two learning algorithms, testing on all combinations of representations and  $\chi^2$  feature size limits. For the feature size limits, we tried values between [400:4000] in steps of 400. The results display the best configuration for each representation.

As shown, the learning algorithms outperformed the baseline for all three metrics, supporting hypothesis H1. This shows that some questions cannot be easily distinguished by simply looking for factoid commands within the questions.

In regards to the second hypothesis, both learning algorithms performed best when using character *n*-grams, especially when restricted by word boundaries, thus supporting H2. We believe this is due to the character *n*-grams being able to better handle noisy nature of chat, especially with the misspellings and abbreviations.

We next examine what *type* of questions do these learning algorithms perform well on, especially when compared to our baseline. For this, we focus on the results of applying SVMs. Figure 4 compares the difference of F<sub>0.5</sub> scores between SVM and the baseline. For most factoids, SVMs performed better or had similar performance to the baseline. The small number of factoids it performed worse on were generally factoids both

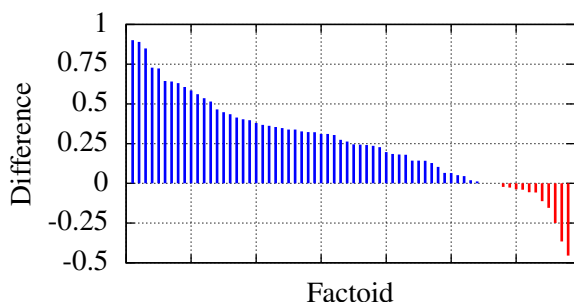


Figure 4: Comparison of  $F_{0.5}$  scores between SVM and baseline.

SVM and the baseline performed poorly, finding low  $F_{0.5}$  scores.

Figure 5 shows the  $F_{0.5}$  scores achieved by SVM for each individual factoid category. These are ordered by their distribution in the corpus (see Figure 3). One fact shown by this figure is that there is not a strong correlation between the distribution size and the result achieved by the SVM. While having more examples within a category does help, there are plenty of factoids where SVM performed poorly. This shows that it is the difficulty of the questions themselves, and not the amount of examples, which causes difficulty for the SVMs, let alone learning algorithms for this domain.

One set of questions SVMs performs well on are questions where users are subsequently directed to another channel. This includes Ubuntu’s non-English channels and channels that provide support for other Linux distributions. SVMs did well on all the non-English factoids, with  $F_{0.5}$  scores ranging between 0.88 (for Chinese) and 0.99 (for Russian). This is probably due to these questions having uncommon features, such as non-English words or software that is not supposed to be discussed in `#ubuntu`.

One similar pair of questions, which are addressed by two factoids, caused some confusion for the learners – asking for permission to ask a question (e.g., “Can I ask a question?”) or asking if anyone can help without stating their problem (e.g., “Can anyone help me?”). This commonly happens with first time visitors to the channel, as they do not know the channel guidelines and will then ask for permission to ask a question or if someone could help them. The channel operators try to encourage users to just ask their question – this happens frequently enough that there are two factoids (labels *ask* and *anyone* in the corpus)

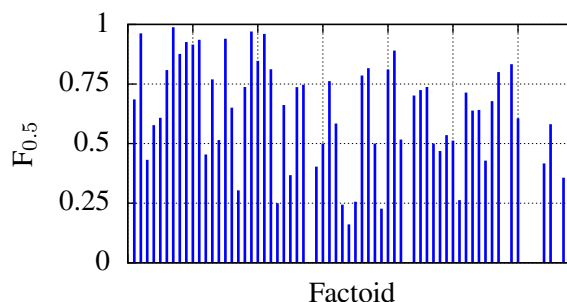


Figure 5:  $F_{0.5}$  scores for each factoid when applying SVM, ordered by distribution.

to answer these similar questions. Unfortunately, there is a lack of consistency in invoking these two factoids, and as such the learning algorithms we tested had difficulty with these questions.

Some other types of questions that SVMs struggle with are those that cover a wide-range of possible questions. For example, `#ubuntu` can be used either in cases to explain what the channel topic is (for those asking), or to get users on topic (with the possible off-topic message types being wide-ranging); *details*, which can be used whenever someone asks a question or for help without providing enough details for anyone to begin to help; and *wine* to help users with problems running any type of Windows program under Linux. These types of questions may require a human to aid in answering, as it would be difficult to learn all possible types of questions that are covered by these factoids.

## 7 Conclusions

We have investigated applying supervised learning algorithms to classify questions as HAQ or BAQ, and our results show that these algorithms can outperform a non-learning baseline approach. We also show that character  $n$ -grams are a better representation than traditional bag-of-words for our task. More importantly, the learning algorithms can answer some types of questions well, indicating that a self-invoking bot can be created that can answer common questions with confidence.

Future work to extend this is to apply unsupervised methods for finding additional questions to match with the factoids. This would greatly extend what we have presented, as we were restricted to the manually-labeled messages to match questions with answers. We plan on applying a case-based reasoning framework (Richter and Weber, 2013) to achieve such a goal.

A final area to investigate is an extension of `ubottu` that can learn to update its knowledge. Currently, only a few users are allowed to edit or add new factoids to `ubottu`. It would be advantageous if it could add new commands and factoids itself by summarizing common answers, or update outdated factoids should it see a common pattern of answers conflicting with its knowledge.

## Acknowledgments

Thanks to NRL for funding this research. David Uthus performed this work while an NRC post-doctoral fellow located at the Naval Research Laboratory. The views and opinions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of NRL or the DoD.

## References

- Christopher D. Berube, Janet M. Hitzeman, Roderick J. Holland, Robert L. Anapol, and Stephen R. Moore. 2007. Supporting chat exploitation in DoD enterprises. In *Proceedings of the International Command and Control Research and Technology Symposium*. CCRP.
- Emily R. Budlong, Sharon M. Walter, and Ozgur Yilmazel. 2009. Recognizing connotative meaning in military chat communications. In *Proceedings of Evolutionary and Bio-Inspired Computation: Theory and Applications III*. SPIE.
- Robin D. Burke, Kristian J. Hammond, Vladimir Kulyukin, Steven L. Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the FAQ FINDER system. *AI Magazine*, 18(2).
- Kevin Dela Rosa and Jeffrey Ellen. 2009. Text classification methodologies applied to micro-text in military chat. In *Proceedings of the International Conference on Machine Learning and Applications*, pages 710–714. IEEE Computer Society.
- Liangjie Hong and Brian D. Davison. 2009. A classification-based approach to question answering in discussion boards. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–178. ACM.
- Charles Lee Isbell, Michael Kearns, Satinder Singh, Christian R. Shelton, Peter Stone, and Dave Korman. 2006. Cobot in LambdaMOO: An adaptive social statistics agent. *Autonomous Agents and Multi-Agent Systems*, 13(3):327–354.
- Jihie Kim, Erin Shaw, Grace Chern, and Donghui Feng. 2007. An intelligent discussion-bot for guiding student interactions in threaded discussions. In *Proceedings of the AAI Spring Symposium on Interaction Challenges for Intelligent Assistants*. AAAI.
- 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.
- Michael M. Richter and Rosina O. Weber. 2013. *Case-Based Reasoning: A Textbook*. Springer Berlin.
- Samira Shaikh, Tomek Strzalkowski, Aaron Broadwell, Jennifer Stromer-Galley, Sarah Taylor, and Nick Webb. 2010. MPC: A multi-party chat corpus for modeling social phenomena in discourse. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 2007–2013. European Language Resources Association.
- David C. Uthus and David W. Aha. 2013a. Multiparty chat analysis: A survey. *Artificial Intelligence*, 199-200:106–121.
- David C. Uthus and David W. Aha. 2013b. The Ubuntu Chat Corpus for multiparty chat analysis. In *Proceedings of the AAI Spring Symposium on Analyzing Microtext*, pages 99–102. AAAI.
- Tom Chao Zhou, Michael R. Lyu, and Irwin King. 2012. A classification-based approach to question routing in community question answering. In *Proceedings of the 21st International Conference Companion on World Wide Web*, pages 783–790. ACM.

# Alignment-based Annotation of Proofreading Texts toward Professional Writing Assistance

Ngan L.T. Nguyen

University of Information Technology  
Hochiminh, Vietnam  
ngannlt@uit.edu.vn

Yusuke Miyao

National Institute of Informatics  
Tokyo, Japan  
yusuke@nii.ac.jp

## Abstract

This work aims at constructing a corpus to satisfy such requirements to support research towards professional writing assistance. Our corpus is a collection of scientific work written by non-native speakers that has been proofread by native English experts. A new annotation scheme, which is based on word-alignments, is then proposed that is used to capture all types of inarticulations and their corrections including both spelling/grammatical error corrections and paraphrases made by proofreaders. The resulting corpus contains 3,485 pairs of original and revised sentences, of which, 2,516 pairs contain at least one articulation.

## 1 Introduction

Detection and correction of misspellings and grammatical errors have been recognized as key techniques for writing assistance, and have extensively been studied in natural language processing (NLP) (Whitelaw et al., 2009; Gamon, 2010; Tetreault et al., 2010; Park and Levy, 2011). However, correcting misspellings and grammatical errors, which can be performed by normal English native speakers, does not satisfy all the requirements of professional writing (Futagi, 2010). The core of the proofreading process, in reality, is paraphrasing inarticulations, which can only be done by expert proofreaders. Considering the two paraphrased sentences (1a) and (1b) below, we can see that sentence (1b) is likely to be considered better by most people (Williams and Colomb, 2010), although neither of them contains any misspellings or grammatical errors.

(1a) *The outsourcing of high-tech work to Asia by corporations means the loss of jobs for many middle-class American workers.*

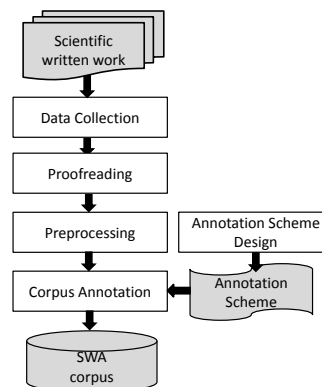


Figure 1: Methodology for corpus annotation

(1b) *Many middle-class American workers are losing their jobs, because corporations are outsourcing their high-tech work to Asia.*

(Williams and Colomb, 2010)

Most of the existing corpora are designed to capture errors in spelling and grammar, but they have not paid enough attention to paraphrasing.

We constructed a corpus that we called scientific writing assistance corpus (SWA), to support research on assistance with scientific-writing that captures all types of inarticulations, including those in both misspellings/grammar and paraphrasing. We have used the term *inarticulation* and *inarticulation correction* instead of *error* and *error correction* in this paper, to include in our task the paraphrasing, which is actually not errors.

Figure 1 overviews the methodology we proposed to construct the corpus. Scientific work written by non-native researchers or graduate students are collected (i.e., data collection, see Section 3), and this was then proofread by English native experts (i.e., proofreading). After that, we preprocessed the documents to convert them into a predefined format (i.e., preprocessing, see Section 3). Annotators with linguistic backgrounds were asked to strictly follow our annotation scheme, which had been designed to capture all types of

inarticulations (i.e., annotation scheme design, see Section 2).

Our corpus construction had several substantial advantages in comparison to the existing corpora such as the NUCLE (Dahlmeier and Ng, 2011), NICT\_JLE (Izumi et al., 2004) and KJ corpora (Nagata et al., 2011). First, the proofreading process is separated from the annotation process. By doing this, both the writer and the proofreader were unaware of the construction of the corpus, so it could capture real articulations and corrections to these. Second, the alignment-based annotation scheme was employed in annotations to capture all types of articulation correction. This allowed us to annotate discontinuous paraphrasing patterns, which were not neatly handled in other corpora. Third, paraphrases were captured, and were proved to be an important type of articulation correction for advanced learners.

The main contributions of this work are in the annotation of paraphrasing and its annotation method, in context of professional proofreading. Statistics for the SWA corpus was given in Section 4). We compared the grammatical errors annotated in the obtained corpus with those in the KJ corpus and NUCLE corpus, two popular corpora often used for research on grammatical error correction (Section 5), and performed an analysis of the paraphrases annotated (Section 6). Our analyses also show the potential of NLP research toward professional text revision.

## 2 Annotation scheme design

We extended the alignment-based paraphrase annotation scheme of Cohn et al. (2008) by categorizing the alignments into more fine-grained types (see Figure 2) to capture all types of inarticulation corrections. Figure 3 outlines example annotations to illustrate our annotation scheme. The alignments at the top level, are divided up into four broad types: Preserved, Metadata, Inarticulation Bi-alignment and Inarticulation Mono-alignment.

The Preserved type of alignments is the most trivial type that connects words with the same surface and function, e.g., *the*, *efficiency*, *various*, *methodologies* in Figure 3(A). Still, there are many cases where two words have the same surface form, but do not have the same functions in the original and the proofread sentences. For instance, the word *of* in the above example appears in both the sentences, but the two occurrences are

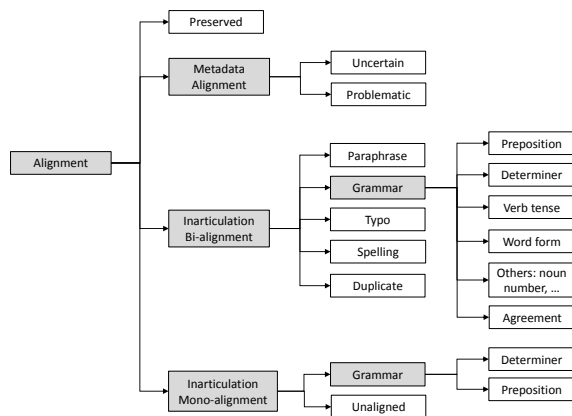


Figure 2: Proposed tagset. Categories in gray are used for classification but not for tagging.

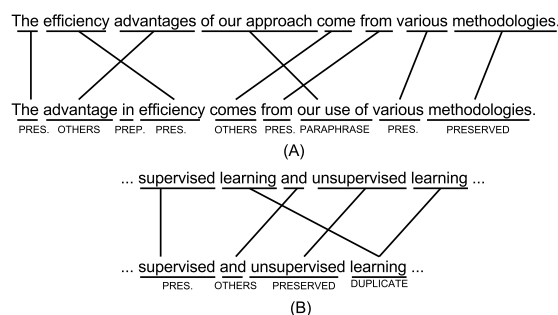


Figure 3: Annotations using our annotation scheme. Top has original texts, and bottom has the proofread text.

not aligned, because they modify different words, i.e., *approach* and *methodologies* in this case.

Inarticulation alignments including mono-alignments and bi-alignments are for capturing inarticulations and their corrections. The Grammar subtype of inarticulation alignments is not used for all types of grammatical errors as in the other annotation scheme, but is limited to some well-defined types of grammatical errors, which will be explained later in Section 2.1. The other subtypes are Duplicate, Spelling, Typo, and Unaligned, which will be explained in the following.

- **Duplicate:** A duplicate alignment connects words that appear once in the original sentence, but more than once in the proofread sentence, or vice versa. This tag captures the correction for articulations like the word *learning* in the example in Figure 3(B).
- **Spelling:** A spelling alignment is used for misspellings, e.g., *occured*→*occurred*<sup>1</sup>. This also includes the use of hyphens, e.g., *state of*

<sup>1</sup>The expression to the right of the arrow (→) is the preferred expression within context of writing



*the art*→*state-of-the-art*.

- **Typo:** The expression typo is a short form of typographical error, which refers to errors caused by typing mistakes. If annotators judge that the error is likely to be caused by a typing mistake, they should mark the errors as typo. Typo may be considered to be less important in writing assistance.
- **Unaligned:** An unaligned mono-alignment is used for words in the original sentence that have no correspondences in the proofread sentence, or vice versa.

Reordering of words are naturally captured by cross alignments, so we do not create a type for this. Punctuation marks are not annotated.

Besides, alignments have additional features to capture information that is specific to proofreading by humans. The current features are: Uncertain and Problematic. An alignment is marked as *uncertain* when the proofreader is not confident in the correction. This type is specific to the proofreading process. When the native proofreader is doubtful about his/her understanding of the original sentence, he/she will comment on it by stating “*I do not understand this,*” or “*This correction is a guess*”. An alignment is classified as *Problematic* when the annotators discover that the proofreader has made an erroneous correction. This happens when the proofreader misunderstands the author’s intention. Although such situations can be rare, this tag is designed to offer a mechanism for annotators to provide feedback.

## 2.1 Grammar

Grammar-typed alignments connect a grammatical error in the original sentence with its correction in the proofread sentence. Grammatical errors in our annotation scheme are comprised of errors with determiners, prepositions, verb tenses, word forms, agreement, and others. They are tagged with the corresponding tags called Determiner, Preposition, Verb tense, Word form, Agreement, and Others. The Others type merges several specific subtypes of grammatical errors, including noun number, verb number, wh-word choice, or conjunction choice. Note that we do not use Others as a catch-all type. Except for Agreement, most of the subtypes of the Grammar type can be aligned well with the error types in the error taxonomies used by the existing corpora. The Agreement type is used to capture the number agree-

ments of articles and nouns, genitives and nouns, or nouns and verbs, when a change in the number of one word forces us to change the number and form of another word.

## 2.2 Paraphrase

Any type of correspondence that cannot be classified into these types above is marked Paraphrase. In other words, Paraphrase is used as a catch-all type. Those errors that require complex corrections, i.e., corrections to phrase structures or sentence structures, which are not classified into the Grammar type, are captured with Paraphrase. We have followed the definition of paraphrases in the guidelines for paraphrase annotation by Callison-Burch et al. (2006): “*paraphrases convey the same meaning but are worded differently*”. We have two rules of thumb for the boundary of paraphrases: (1) shorter paraphrases are preferable (similar to (Callison-Burch et al., 2006)), and (2) a paraphrase alignment should not contain an alignment of other types in it.

## 3 Data collection and preprocessing

We collected scientific works that were written by seven authors with two language backgrounds Japanese and Vietnamese. The collected documents included different types of scientific publications such as short papers, full papers, and book chapters. We will use the terminology *document* to refer to a written work of any type. The collected documents belonged to two domains or fields of studies, which were computer vision (11 documents) and natural language processing (7 documents); and all were proofread by native English experts.

We then preprocessed these documents to convert them into a standard format. Non-text information such as figures and tables were removed. Format tags such as LaTeX’s tags were also removed. We separated the original text and the proofread text for each document, and aligned the sentences in these two texts, so that a line in the original text corresponded to a line in the proofread text. We found that there were cases where a sentence in the original text should have been aligned with more than one sentence in the proofread text or vice versa. We allowed two or more sentences to be aligned in such cases.



## 4 Corpus annotation and results

We made use of Yawat, a web-based word-alignment annotation tool (Germann, 2008) to annotate the corpus. Yawat accepts text files containing pairs of aligned sentences as input. We applied a simple string-matching algorithm to produce default Preserved and Unaligned alignments for the corpus to save annotation time and effort.

The corpus was annotated by two annotators with linguistics background. The agreement between them was measured using the F1-score formula similarly to that by Cohn et al. (2008). *Atom-alignments*, or one-to-one alignments, were generated from the bi- and mono-alignments. An  $M \times N$  multiple alignment would result in  $M \times N$  atom alignments. We removed the preserved atom-alignments that were annotated by both annotators, because they occupied the majority of atom alignments but were not a meaningful indicator of inter-annotator agreement. Considering annotations by one annotator as gold annotations, we then calculated recall, precision, and F1-score over all the annotated alignments in the two versions of the SWA corpus. The overall F-scores with and without considering alignment types were 0.637 and 0.716 respectively. It can be seen that our inter-annotator agreement measure without considering alignment classification is comparable to those reported by Cohn et al. (2008) (0.71, 0.74, and 0.76, for the three datasets of MTC, Leagues, and News, respectively). This is reasonable because when alignment classification is taken into account, the annotation task is more difficult, so the inter-annotator agreement is lower.

A total of 4,686 Inarticulation alignments were annotated for 2,516 pairs of sentences in 18 documents. 69,738 (91.8%) of the total of 75,968 words in the corpus were annotated with Preserved alignments. Table 1 lists the ratios (%) of broad types of alignments. We can see that the Grammar errors, both in bi- and mono-alignments, occupy 58.1% of the total errors, which is not a surprise. Paraphrase alignments occupy a significant part, i.e., 29.3% of the total. These figures indicate that paraphrasing is an essential type for scientific writing; therefore, research on writing assistance should pay more attention to error correction by using paraphrasing.

The ratios of the subtypes of Grammar alignments are listed in the column named SWA (the

Alignment Type	Count	Ratio (%)
Paraphrase	1,372	29.3
Bi-Grammar	1,511	32.2
Typo	68	1.5
Spelling	308	6.6
Duplicate	13	0.3
Preserved	2	0.0
Mono-Grammar	1,212	25.9
Unaligned	200	4.3
TOTAL	4,686	100.0

Table 1: Statistics for all alignments (except for the Preserved type) annotated in the corpus

name of our corpus) in Table 2. Out of all grammatical errors, determiners caused a lot of troubles for non-native writers from the Japanese and Vietnamese language backgrounds, even though the authors of the collected documents all had an advanced level of proficiency in English. This may be because of the difference between the characteristics of their background languages and the English language.

## 5 Cross-corpora comparison for grammatical errors

This section compares the grammatical-error annotations (Grammar alignments) in our corpus with those annotated in the KJ and NUCLE corpora. The Grammar types of errors in our scheme are restricted to well-defined types of grammatical errors. It would be interesting to analyze the differences in grammatical errors made by writers of the three corpora. The writers for our SWA corpus were graduate students and researchers in the field of computer vision and natural language processing, who could be considered to be advanced learners. The writers for the KJ and NUCLE were Japanese and Singaporean students, respectively.

As the three corpora used different annotation schemes, we created a mapping between compatible tags in the three tagsets to compare our corpus with theirs. This mapping is summarized in Table 3. The annotation scheme used for the KJ corpus, called KJ annotation scheme, was a simplified version of the NICT\_JLE annotation scheme (Nagata et al., 2011). The definitions of types and marking schemes are basically similar in the two annotation schemes, but the KJ annotation scheme merges several subtypes into one type, for example, the

Type	KJ Count ( $\times\alpha$ )	KJ (%)	NUCLE Count ( $\times\beta$ )	NUCLE (%)	SWA Count	SWA (%)
Determiner	543 (726)	18.7	6,004 (641)	12.9	1,176	25.1
Preposition	377 (504)	13.0	7,312 (781)	15.7	547	11.7
Others	404 (540)	13.9	5,486 (543)	10.9	427	9.1
Verb tense	249 (333)	8.6	3,288 (351)	7.1	369	7.9
Word form	317 (423)	10.9	2,241 (239)	4.8	151	3.2
Agreement	146 (195)	5.0	1,578 (168)	3.4	53	1.1
TOTAL of Grammar	2,036 (2,723)	70.0	25,509 (2,723)	54.7	2,723	58.1
Total of all types	2,907	100.0	46,597	100.0	4,686	100.0

Table 2: Statistics for Grammar alignments in SWA in comparison with KJ corpus and NUCLE corpus with  $\alpha = \text{TOTAL}_{SWA} / \text{TOTAL}_{KJ}$ ,  $\beta = \text{TOTAL}_{SWA} / \text{TOTAL}_{NUCLE}$

KJ	NUCLE	SWA
at	ArtOrDet	Determiner
prp	Wcip	Preposition
n_num, rel	Nn, Vform	Others
v_tns	Vt	Verb tense
aj, v_lxc	Wform	Word form
v_agr	SVA	Agreement

Table 3: Tagset mapping of KJ, NUCLE, and SWA for comparison. Note that only corresponding tags are mapped.

*noun inflection*, *noun case*, *noun countability* and *complement of noun* of the NICT\_JLE annotation scheme, are merged into one type, the *noun lexical*. The KJ tagset contains 19 tags, fewer than the total number of 45 error tags in the NICT\_JLE tagset (Izumi et al., 2004). The NUCLE tagset has more fine-grained tags than the KJ tag set (27 tags).

The four types Determiner, Preposition, Verb tense, and Agreement in our tagset have counterparts in the KJ tagset, which are *at* (article), *prp* (preposition), *v\_tns* (verb tense), and *v\_agr* (verb agreement) tags, and in the NUCLE tagset, which are *ArtOrDet* (article or determiner), *Wcip* (wrong collocation/idiom/preposition), *Vt* (verb tense), and *SVA* (subject-verb agreement). Note that subject-verb agreement is only part of the Agreement type in our annotation scheme (see Section 2). The counts for the Others type were sums of the *n\_num* (noun number) and *rel* (relative) types for the KJ corpus, and of the *Nn* (noun number) and *Vform* (verb form) types for the NUCLE corpus. The Word-form figure of the KJ corpus was a sum of the *aj* (adjective), and *v\_lxc* (verb lexical) types. As NUCLE has the exactly corresponding type called *Wform* (word form), so we used the count of this type in our comparison.

The comparison statistics are summarized in Ta-

ble 2. We can see in this table that the ratios (%) of the totals of basic grammatical types over the totals of all annotated inarticulations, are significantly different for the three corpora, which correspond to 70.0%, 54.7%, and 58.1% for KJ, NUCLE, and SWA. The differences probably reflect the actual proficiency levels of the writers. Texts in KJ and NUCLE are written by college students, but they are not the same. This can be explained by the fact that NUCLE’s college students are studying in Singapore, where English is used as an official language, while KJ’s students are living in Japan, where English is not usually heard in daily life. The SWA’s writers are also not living in an English-speaking environment, but they made fewer basic grammatical errors than KJ’s students, which is reasonable because they have a higher proficiency level.

We normalized the count of each error type by using  $\alpha$  and  $\beta$  listed in Table 2 to directly compare the three corpora in more detail. The normalized counts are in parentheses, next to the actual counts. To our surprise, the SWA’s writers, who were scientific writers, make numerous determiner errors: 1,176 errors, compared to 726 (KJ) and 641 (NUCLE). KJ’s students made fewer errors of this type than SWA’s writers. This is possibly due to the difference in the complexity of the sentence structures used by the three groups of writers. KJ’s students wrote very short sentences, while advanced learners tended to write those that were longer and more complex. Additional analyses of the sentence lengths and structures would clarify this further.

## 6 Analysis for paraphrase alignments

We carried out an analysis of the annotated Paraphrase alignments for understanding the challenges and possible solutions for research toward automatic proofreading (Table 4). For this anal-

Type	Examples of annotation	Count	%
1.Short-form ↔ Long-form	<i>PCA</i> → <i>principle component analysis</i>	2	0.6
2. Verb ↔ Prepositional phrase	<i>to collect</i> → <i>of collecting</i>	13	3.6
3.Relative clause ↔ Participle	<i>needed</i> → <i>that need</i>	5	1.4
4.Active ↔ Passive	<i>has not ... studied</i> → <i>has not ... been ... studied</i>	13	3.6
5.Anaphoric pronoun ↔ Referent	<i>this</i> → <i>the result</i>	22	6.1
6.Selection	<i>have</i> → <i>provide</i> <i>on the contrary</i> → <i>on the other hand</i> <i>frontal</i> → <i>the front of</i>	131	36.4
7.Mis-use/ Ad- dition	<i>good point</i> → <i>advantage</i>	55	15.3
8.Unknown/ Simplification	<i>It is better if ... are used</i> → <i>Using ... is better</i>	32	8.9
9.Complex		87	24.2
TOTAL		360	100.0

Table 4: Subtypes of Paraphrase alignments representing different confusing patterns of writers.

ysis, we randomly picked 20 Paraphrase alignments from each annotated document, and manually categorized them into nine subtypes that approximately represented different confusing patterns of writers.

The first five subtypes in the table are rather well-defined types. They were used for such transformations as between short- and long-form of acronyms, or relative clauses and their reduced forms, and so on. These well-defined paraphrases occupy 20.8 percent of all the total samples. The transformation between active and passive forms, and between anaphoric pronouns and their concrete forms could be challenging, because it requires correct interpretation of the event or entity being mentioned. For not-well-defined paraphrases, we classified them based on the number and the part-of-speech of the inclusive words.

The Selection subtype is for the replacement of a word with another word of the same part-of-speech, or an idiom with another idiom. There were several causes of this type of inarticulation. One cause was that the writers used less-formal or ambiguous words, which were inappropriate for scientific writing style. Another cause was that they selected a word which did not precisely describe the intended meaning, due to the interference by the writer’s background-language or other reasons. The latter reason would be more challenging for automatic proofreading applications.

Selection-typed paraphrasing is very important for writing assistance, not only because of its frequency but also it is a representative example of the increasing fluency of texts.

The Mis-use/Addition subtype is applied when a word in the original text is replaced with a sequence of several words in the proofread text. This often happens when the original words do not provide enough details, or mis-describe what the writers mean. Unknown/Simplification is the reverse subtype of Mis-use/Addition. This subtype indicated that non-native writers sometimes used long descriptions instead of compact words, such as *good point* instead of *advantage*. These two subtypes reveal the demand for techniques to simplify, or to provide more information to the text.

The Complex subtype is for many-to-many alignments. While the changes made by the other subtypes above are rather local, this subtype often required global changes at high levels of a sentence structure, such as those in the example *it is better if ... are used* → *using ... is better*. Previous studies on text revision have suggested that such changes are necessary for the coherence of a bigger discourse such as a paragraph or whole document (Williams and Colomb, 2010). How to make use of discourse information in automatic proofreading is an interesting issue of NLP studies using our corpus.

## 7 Conclusion

We described the SWA corpus, which was constructed to support studies on assistance techniques for professional writing. The traditional problem of error annotation was reformulated as a paraphrase annotation of pairs of the original and proofread sentences. This view inspired us to extend the alignment-based annotation scheme to be used for our annotation process. The comparison with two existing popular corpora revealed that grammatical errors made by different types of writers varied a great deal. The advanced writers tended to make more inarticulations that require paraphrasing.

The SWA corpus can be used as benchmark data for different tasks including grammatical error correction, paraphrase extraction, and automatic alignment, in context of proofreading. Further research should be carried out for paraphrasing techniques. The corpus is made available for research community on request basis.

## References

- Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *J. Artif. Int. Res.*, 38(1):135–187, May.
- Chris Callison-Burch, Trevor Cohn, and Mirella Lapata. 2006. Annotation guidelines for paraphrase alignment, 12.
- Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Comput. Linguist.*, 34(4):597–614, December.
- Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 915–923, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Robert Dale and Adam Kilgarriff. 2010. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, Dublin, Ireland.
- Robert Dale and Adam Kilgarriff. 2011. Helping our own: Text massaging for computational linguistics as a new shared task. In *Proceedings of the International Natural Language Generation Conference 2011*, Nancy, France, September.
- Yoko Futagi. 2010. The effects of learner errors on the development of a collocation detection tool. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10, pages 27–34, New York, NY, USA. ACM.
- Michael Gamon. 2010. Using mostly native data to correct errors in learners' writing: a meta-classifier approach. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 163–171, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ulrich Germann. 2008. Yawat: yet another word alignment tool. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*, HLT-Demonstrations '08, pages 20–23, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Emi Izumi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2004. Sst speech corpus of japanese learners' english and automatic detection of learners' errors. 28:31–48.
- John Milton and Vivying S. Y. Cheng. 2010. A toolkit to assist l2 learners become independent writers. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, CL&W '10, pages 33–41, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ryo Nagata, Edward Whittaker, and Vera Sheinman. 2011. Creating a manually error-tagged and shallow-parsed learner corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1210–1219, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Y. Albert Park and Roger Levy. 2011. Automated whole sentence grammar correction using a noisy channel model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 934–944, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Benno Stein, Martin Potthast, and Martin Trenkmann. 2010. Retrieving customary web language to assist writers. In *Proceedings of the 32nd European conference on Advances in Information Retrieval*, ECIR'2010, pages 631–635, Berlin, Heidelberg. Springer-Verlag.
- Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*.
- Casey Whitelaw, Ben Hutchinson, Grace Y. Chung, and Gerard Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 890–899, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Joseph M. Williams and Gregory G. Colomb. 2010. *Style: Lessons in clarity and grace*. Boston, MA: Longman.

# Toward Automatic Processing of English Metalanguage

Shomir Wilson\*

School of Informatics  
University of Edinburgh  
10 Crichton Street  
Edinburgh EH8 9AB, UK

School of Computer Science  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213 USA

shomir@cs.cmu.edu

## Abstract

The metalinguistic facilities of natural language are crucial to our ability to communicate, but the patterns behind the appearance of metalanguage—and thus the clues for how we may instruct computers to detect it—have remained relatively unknown. This paper describes the first results on the feasibility of automatically identifying metalanguage in English text. A core metalinguistic vocabulary has been identified, supporting intuitions about the phenomenon and aiding in its detection and delineation. These results open the door to applications that can extract the direct, salient information that metalanguage encodes.

## 1 Introduction

In linguistic communication it is sometimes necessary to refer to features of language, such as orthography, vocabulary, structure, pragmatics, or meaning. *Metalanguage* enables a speaker to select a linguistically-relevant referent over (or in addition to) other typical referents (Audi, 1995). Metalanguage is illustrated in sentences such as

- (1) *Graupel* refers to a kind of precipitation.
- (2) The name is actually *Rolla*.
- (3) *Keep tabs on* is a colloquial phrase.
- (4) He wrote “**All gone**” and nothing more.

The roles of the bold substrings in the above sentences contrast with those in (5)-(8) below:

- (5) **Graupel** fell on the weary hikers.
- (6) **Rolla** is a small town.
- (7) **Keep tabs on** him, will you?
- (8) They were **all gone** when I returned.

Conventional stylistic cues, such as italics in (1), (2), and (3) and quotation marks in (4), sometimes help the audience to recognize metalinguistic statements in written language. In spoken language or in written contexts where stylistic cues are not used, the audience is expected to identify metalinguistic statements using paralinguistic cues (such as intonation, when speaking) or context and meaning.

Metalanguage is both pervasive and, paradoxically, the subject of limited attention in research on language technologies. The ability to produce and understand metalanguage is a core linguistic competence that allows humans to converse flexibly, unrestricted by domain (Anderson et al., 2002). Humans use it to establish grounding, verify audience understanding, and maintain communication channels (Anderson et al., 2004). Metalanguage encodes direct and salient information about language, but many typical examples thwart parsers with novel word usage or arrangement (Wilson, 2011a). Metalanguage is difficult to classify through the interpretive lens of word senses, given that conventional word senses have little relevance when a word appears chiefly “as a word”. The roles of metalanguage in L2 language acquisition (Hu, 2010), expression of sentiment toward others’ utterances (Jaworski et al., 2004), and irony (Sperber and Wilson, 1981) have also been noted.

This paper describes the results of the first effort to automatically identify instances of metalanguage in English text. *Mentioned language*, a common variety of metalanguage, is focused upon for its explicit, direct nature, which makes its structure and meaning easily accessible once an instance is identified. Section 2 reviews a prior project by Wilson (2012) to create a corpus of instances of metalanguage, a necessary resource for the present effort. Section 3 describes an approach to distinguishing sentences that contain

---

\* This research was performed during a prior affiliation with the University of Maryland at College Park.

metalanguage from those that do not, a task referred to as *detection* for brevity. Results show that the performance of this approach roughly matches an implied performance ceiling of inter-annotator agreement. Section 4 describes an approach to *delineate* sequences of words that are directly mentioned by a metalinguistic statement; although the results are preliminary, its accuracy shows promise for future development. Together, these results on detection and delineation show the feasibility of enabling language technologies to extract the salient information about language that metalanguage contains.

## 2 Background

The reader is likely to be familiar with the concept of metalanguage, but a discussion is appropriate to ground the concept and connect to previous work. Section 2.1 summarizes a prior study (Wilson, 2012) to collect instances of metalanguage, and 2.2 reviews some related efforts.

### 2.1 Prior Work

A diverse variety of phenomena in natural language satisfy the intuitive criteria that we associate with metalanguage. The prior study focused on identifying sentences that contained *mentioned language*, a phenomenon defined below:

*Definition: For T a token or a set of tokens in a sentence, if T is produced to draw attention to a property of the token T or the type of T, then T is an instance of mentioned language.*<sup>1</sup>

Here, a *token* is an instantiation of a linguistic entity (e.g., a letter, symbol, sound, word, phrase, or other related entity), and a *property* is an ostension of language (García-Carpintero, 2004; Saka, 2006), such as spelling, pronunciation, meaning (for a variety of interpretations of *meaning*), structure, connotation, or quotative source. Generally attention is drawn to the *type* of T (for example, in Sentences (1)-(4)), but it can be drawn to the *token* of T for self-reference, as in Sentence (9):

(9) “The” appears between quote marks.

Although constructions like (9) are unusual and carry less practical value, the definition accommodates them for completeness.

Mentioned language is a common form of metalanguage, used to perform the full variety of

language tasks discussed in the introduction. However, other metalinguistic constructions draw attention to tokens *outside of* the referring sentence. Some examples of this are (10)-(12) below. Supporting contexts are not shown for these sentences, though such contexts are easily imagined:

- (10) Disregard the last thing I said.
- (11) That spelling, is it correct?
- (12) People don’t use those words lightly.

In each of the above three sentences, a linguistic entity (an utterance, a sequence of letters, and a sequence of words, respectively) is referred to, but the referent is contained in a separate sentence. The referent may have been produced by a different utterer or appeared in a different medium (e.g., speaking aloud while referring to written text). These “extra-sentential” forms of metalanguage have clear value to understanding discourse and coreference. The focus on mentioned language is a limitation to the present work, to utilize an existing corpus and to apply tractable boundaries to the identification tasks.

The mentioned language corpus of the prior study<sup>2</sup> was constructed by filtering a large volume of sentences with a heuristic, followed by annotation by a human reader. A randomly selected subset of articles from English Wikipedia was chosen as a source for text because of its representation of a large sample of English writers (Adler et al., 2008), the rich frequency of mentioned language in its text, and the frequent use of stylistic cues in its text that delimit mentioned language (i.e., bold text, italic text, and quotation marks). Sentences were sought that contained at least one of these stylistic cues and a *mention-significant* word in close proximity. Mention-significant words were a set of 8,735 words and collocations with potential metalinguistic significance (e.g., *word*, *symbol*, *call*), extracted from the WordNet lexical ontology (Fellbaum, 1998). Phrases highlighted by the stylistic cues were considered *candidate instances*, and these were labeled by a human reader, who determined that 629 sentences were *mention sentences* (i.e., containing instances of mentioned language) and the remaining 1,764 were not. Mention sentences were categorized based on functional properties that emerged during categorization. Table 1 shows some examples of collected mention sentences in each category.

<sup>1</sup> This definition was introduced by Wilson (2011a) along with a practical rubric for evaluating candidate sentences. For brevity, its full justification is not reproduced here.

<sup>2</sup> The corpus is available at [http://www.cs.cmu.edu/~shomir/um\\_corpus.html](http://www.cs.cmu.edu/~shomir/um_corpus.html).

Category	Examples
Words as Words	The IP Multimedia Subsystem architecture uses the term <b>transport plane</b> to describe a function roughly equivalent to the routing control plane. The material was a heavy canvas known as <b>duck</b> , and the brothers began making work pants and shirts out of the strong material.
Names as Names	<b>Digeri</b> is the name of a Thracian tribe mentioned by Pliny the Elder, in The Natural History. Hazrat Syed Jalaluddin Bukhari's descendants are also called <b>Naqvi al-Bukhari</b> .
Spelling or Pronunciation	The French changed the spelling to <b>bataillon</b> , whereupon it directly entered into German. Welles insisted on pronouncing the word apostles with a hard <b>t</b> .
Other Mentioned Language	He kneels over Fil, and seeing that his eyes are open whispers: <b>brother</b> . During Christmas 1941, she typed <b>The end</b> on the last page of Laura.

Table 1: Examples of mentioned language from the corpus. Instances of the phenomenon appear in bold, with the original stylistic cues removed.

To verify the reliability of the corpus and the definition of mentioned language, three additional expert annotators independently labeled a shuffled set of 100 sentences, consisting of 54 randomly selected mention sentences and 46 randomly selected non-mention sentences. All three agreed with the primary annotator on 46 mention sentences and 30 non-mention sentences, with an average pairwise Kappa of 0.74. Kappa between the primary annotator and a hypothetical “majority voter” of the additional annotators was 0.90. These results were seen as a moderate indication of reliability and a potential performance ceiling for automatic identification.

## 2.2 Related Work

The present effort is believed to be the first to automatically identify a natural variety of metalanguage in English text. Aside from the corpus described above, the only other significant corpus of metalanguage was created by Anderson et al. (2004), who collected metalinguistic utteranc-

es in conversational English. A lack of phrase-level annotations in their corpus as well as substantial noise made it suboptimal for the present effort. However, it is possible (if not likely) that indicators of metalanguage differ between written and spoken English, lending importance to the Anderson corpus as a resource.

Metalanguage has a long history of theoretical treatments, which chiefly explained the mechanics of selected examples of the phenomenon. Many addressed it through the related topic of *quotation* (Cappelen and Lepore, 1997; Davidson, 1979; Maier, 2007; Quine, 1940; Tarski, 1933), and others previously cited in this paper discussed it directly as *metalanguage* or the *use-mention distinction*. The definition of mentioned language in Section 2.1 was a synthesis of the most empirically-compatible theoretical treatments, and the present effort to automatically identify metalanguage builds on that synthesis.

## 3 Detection of Mentioned Language

The corpus-building effort used a heuristic to accelerate the collection of mentioned language, but its low precision is impractical for automatic identification. Moreover, the stylistic cues that the heuristic relied upon are often inconsistently applied (or entirely absent in informal contexts), and they are sometimes unavailable for the writer to use or for the audience to extract. This section presents an approach to the *detection* task, to discriminate between mention and non-mention sentences. Early examination of the corpus suggested that mention sentences tend not to have distinct structural differences from non-mention sentences, so a lexical approach was first taken, although combinations of lexical and structural approaches are later explored indirectly through the delineation task. In this section a sentence is assumed to be a sequence of words without stylistic cues for mentioned language.

### 3.1 Approach

To establish performance baselines, a matrix of feature sets and classifiers was run on the corpus with ten-fold cross validation. The feature sets were bags of the following: stemmed words (SW), unstemmed words (UW), stemmed words plus stemmed bigrams (SWSB), and unstemmed words plus unstemmed bigrams (UWUB). Classifiers were chosen to reflect a variety of approaches to supervised learning; as implemented in Weka (Hall et al., 2009), these were Naive Bayes (John and Langley, 1995), SMO (Keerthi

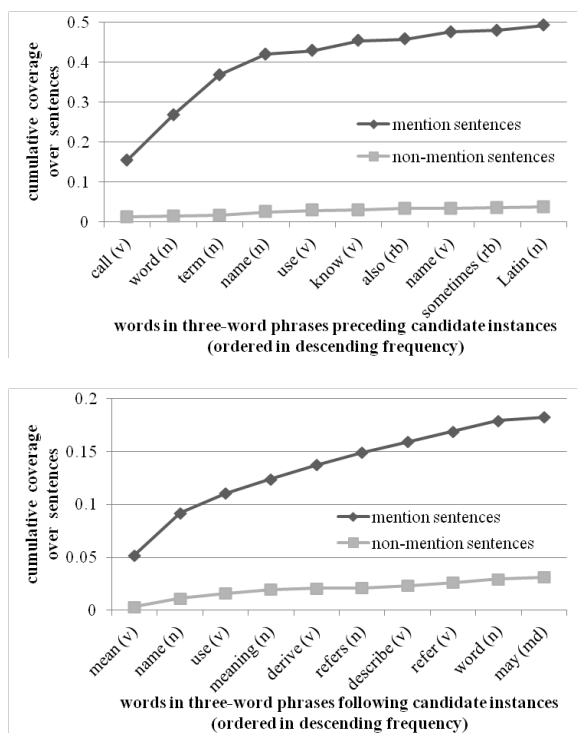


Figure 1. Cumulative coverage over sentences by the most common words before (top) and after (bottom) instances of mentioned language.

et al., 2001), J48 (Quinlan, 1993), IBk (Aha and Kibler, 1991), and Decision Table (Kohavi, 1995).

Prior observations suggested that a small set of approximately ten words significant to metalanguage (“metawords”, informally) appear near most instances of mentioned language (Wilson, 2011b). The metalanguage corpus described in Section 2 provided an opportunity to explore this observation. The sentences in the corpus were part-of-speech tagged and stemmed (using NLTK (Bird, 2006)). Sets were collected of all unique (stemmed) words in the three-word phrases directly preceding and following candidate instances, and (respective of position) these were ranked by frequency. The appearance or non-appearance of these words was then determined over all mention and non-mention sentences. Figure 1 shows the cumulative coverage (i.e., appearance at least once) over sentences for the top ten words appearing before and after candidate instances. For example, *call*, *word*, or *term* were the three most common words before candidate instances; they appear at least once in 36% of mention sentences, but they appear in only 1.6% of non-mention sentences.

The high frequencies of intuitive metawords, combined with their difference in coverage over

mention and non-mention sentences, informed the approach taken to the detection task. To attempt to improve over the baseline, the SW feature set was ranked by information gain, and all features except the top ten were discarded to create the metawords feature set (MW). Feature selection was done using the training set for each cross-validation fold, and the testing data for each fold was pruned correspondingly. The five selected classifiers were then applied to the data.

### 3.2 Results and Discussion

The combination of five feature sets and five classifiers produced 25 sets of annotations, for which precision, recall and F1 were calculated for detecting mentioned language. For brevity, we present the highlights and contrast the metawords approach with baseline performances.

Table 2 compares classifier performances using MW with SW, its closest relation. MW produced improvement for all classifiers except Naive Bayes<sup>3</sup>. The J48-MW combination had the highest F1 and recall of any feature set-classifier combination, though some combinations exceeded its precision. For all feature set-classifier pairs, precision was higher than recall, by as little as 0.024 (IBK-MW) and as much as 0.22 (Decision Table-UW). For the baseline feature sets, the best classifier was consistently SMO, with F1 scores of 0.70, 0.70, 0.73, and 0.71 for SW, UW, SWSB, and UWUB, respectively. J48 was consistently the second best, with F1 scores within 0.01 of SMO for each feature set.

Table 3 lists differences between F1 scores using the MW feature set and each baseline feature set. MW resulted in improvements over the baseline feature sets for nearly all classifiers, and statistically significant improvements (using one-tailed T-tests across the populations of validation folds,  $p < 0.05$ ) were observed for eleven of the sixteen combinations. IBk appeared to benefit the most, with significant improvements over all baseline feature sets, and Naive Bayes the least. In general, recall benefited more than precision.

Examining the MW features confirmed that most were intuitive metawords. Nine words appeared in all ten folds of MW: *name*, *word*, *call*, *term*, *mean*, *refer*, *use*, *derive*, and *Latin*. The last two words are perhaps artifacts of the encyclopedic nature of the source text, but the rest generalize easily. Future research using additional

<sup>3</sup> It seems likely that the method used to create the MW feature set aggravated the Naive Bayes assumption of feature independence.



Classifier	Precision	Recall	F1
Naive Bayes	.76 / <b>.75</b>	.63 / <b>.60</b>	.69 / <b>.66</b>
SMO	.74 / <b>.75</b>	.67 / <b>.70</b>	.70 / <b>.73</b>
IBk	.69 / <b>.74</b>	.64 / <b>.72</b>	.66 / <b>.73</b>
Decision Table	.76 / <b>.74</b>	.61 / <b>.68</b>	.67 / <b>.71</b>
J48	.72 / <b>.75</b>	.69 / <b>.73</b>	.70 / <b>.74</b>

Table 2. The performances of classifiers using the SW and MW (in bold) feature sets.

Classifier	SW	UW	SWSB	UWUB
Naive Bayes	-.024	-.018	.005	.007
SMO	<b>.023*</b>	<b>.026*</b>	.000	.015
IBk	<b>.067*</b>	<b>.088*</b>	<b>.07*</b>	<b>.108*</b>
Decision Table	<b>.038*</b>	<b>.047*</b>	.027	<b>.052*</b>
J48	<b>.037*</b>	<b>.046*</b>	.025	.034

Table 3. Differences between F1 scores from using the MW feature set and baseline feature sets. Statistically significant improvements are starred.

text sources will be necessary to fully verify whether the MW approach and these specific metalinguistic terms are widely applicable.

It also appears that 20% to 30% of instances of mentioned language resist identification using word and bigram-based features alone. Many of the false negatives from this experiment appeared to lack the common metawords that the detection approach relied upon. The sentences below (taken from the corpus) illustrate this lack:

(13) Other common insulting modifiers include “dog”, “filthy”, etc.

(14) To note, in the original version the lyrics read “Jim crack corn”.

While *modifier* in (13) and *read* in (14) have intuitive metalinguistic value, they also have common non-metalinguistic senses. This suggests that an approach incorporating word senses may further improve upon the MW performances, and such an approach is preliminarily explored in the following section.

Finally, it is notable that the best MW performances approach the Kappa score observed between the additional annotators. Although this is an indication of some success, the higher “majority vote” Kappa score of 0.90 remains a meaningful goal for future research efforts.

#### 4 Toward Delineation

After identifying a mention sentence, the task remains to determine the specific sequence of

words subject to direct reference (e.g., the bold words in Sentences (1) through (4) and in other examples throughout this paper). This task, in addition to detection, is necessary to ascribe the information encoded in a metalinguistic statement to a specific linguistic entity.

#### 4.1 Approach

Manual examination of the corpus showed two frequent relationship patterns between metawords and mentioned language. The first was *noun apposition*, in constructions like (15) and (16), where the metaword-noun appears in italics and the mentioned word in bold:

(15) The *term* **auntie** was used depreciatively.

(16) It comes from the root *word* **conficere**.

The second pattern was the appearance of mentioned language in the *semantic role of a meta-word-verb*, as in (17) below:

(17) We sometimes *call* it **the alpha profile**.

Notably these patterns do not guarantee the correct delineation of mentioned language, but their applicability made them suitable for the task.

To assess the applicability of phrase structures and semantic roles to the automatic delineation of mentioned language, case studies were performed on the sets of sentences in the corpus containing the nouns *term* and *word* and the verb *call*. All sentences containing these three metawords (appearing as their respective targeted parts of speech) were examined, including those that did not contain mentioned language, since it was believed that methods of delineation could indirectly perform detection as well. Because of the limited data available, formal experiments were not possible, although the results still have illustrative value.

The noun apposition pattern described above was formalized for *term* and *word* using TRegex search strings (Levy and Andrew, 2006). The 91 sentences in the corpus containing *term* and the 107 containing *word* were parsed using the Stanford Parser (Marneffe et al., 2006), and the TRegex strings were applied to each sentence; when a match occurred, the result was a prediction that a specific sequence of words was mentioned language (delineation), as well as a prediction that the sentence contained mentioned language (detection). The semantic role pattern for *call* was explored similarly using the Illinois Semantic Role Labeler (SRL) (Punyakanok et al., 2008). Each of the 158 sentences in the corpus containing *call* as a verb was processed by

SRL, and when the output contained the appropriate semantic role (i.e., SRL’s “attribute of arg1”) with respect to the metaword, the phrase fulfilling that role was considered a predicted delineation of mentioned language. By proxy, such matching also implied a prediction that the phenomenon was present in the sentence.

## 4.2 Results and Discussion

Delineation was evaluated with respect to the correctness of *label scope*: that is, for a sentence that contained an instance of mentioned language, whether the predicted word sequence exactly matched the sequence labeled in the corpus, overlabeled it (i.e., included the instance of mentioned language plus additional words), or underlabeled it (i.e., did not include the entire instance). To avoid confounding detection and delineation, the statistics on label scope do not include instances when the appropriate pattern failed to annotate *any* phrase in a sentence that contained mentioned language, or annotated a phrase when no mentioned language was present. Such instances are instead represented through *pattern applicability* statistics: when one of the sought relationships between a chosen metaword and a phrase appeared in a sentence, it was considered a positive prediction of the presence of mentioned language. Table 4 shows performance metrics from each of the three case studies.

Noun apposition with either *term* or *word* appeared to be adept at predicting scope, with perfect labels for 97% and 89% of instances, respectively. The instances of overlabeling and underlabeling for these two were mostly due to parsing errors, which occurred prior to applying the TRegex pattern. Overlabeling was a greater problem for *call*, for which 80% of labels were perfect and nearly the rest were overlabeled. Manual examination revealed that the prediction often would “spill” far past the actual end of mentioned language, due to the boundaries of the semantic role in SRL’s output. For example, the entire phrase in bold in (18) below was erroneously predicted to be mentioned language, instead of simply *snow-eaters*:

(18) Winds of this type are called ***snow-eaters*** for their ability to make snow melt or sublime rapidly.

Re-examining the detection task through pattern applicability, noun appositions with *term* and *word* exhibited perfect precision. The false negatives that lowered the recall were again mostly due to parse errors. Precision and recall

Metaword	Label Scope		
	Overlabeled	Underlabeled	Exact
<i>term</i> (n)	0	2	57
<i>word</i> (n)	3	4	57
<i>call</i> (v)	16	1	68

Metaword	Pattern Applicability		
	Precision	Recall	F1
<i>term</i> (n)	1.0	0.89	0.90
<i>word</i> (n)	1.0	0.94	0.97
<i>call</i> (v)	0.87	0.76	0.81

Table 4: Performance statistics for delineation (in the form of label scope) and detection (pattern applicability) for the case studies.

for *call* suffered from two sources of errors: incorrect applications of the semantic role and applications of it that, while valid, did not involve mentioned language.

For the selected metawords, it appeared patterns in noun apposition and semantic roles were moderately effective at delineating as well as detecting mentioned language. However, the accuracy of these patterns was a reflection of the dependability of the underlying language tools, and the case studies in aggregate covered only 33% of the sentences containing mentioned language in the corpus. To create a comprehensive method for delineation, more relationships must be identified between metawords and mentioned language. A perusal of the corpus suggests that these patterns are small in variety but large in quantity: metawords are diverse, and some have non-metalinguistic senses that must be accounted for, as shown by Sentences (13) and (14) and others that resisted detection.

## 5 Conclusion

The detection and delineation methods presented in this paper demonstrate the feasibility of identifying metalanguage in English text. The next goals of this project will be to assimilate metalanguage from additional text sources and integrate the detection and delineation tasks. This will improve performance and provide a richer structural knowledge of metalanguage, which will enable practical systems to incorporate processing of the phenomenon and exploit the linguistic information that it encodes.

## References

- Adler, B. T., de Alfaro, L., Pye, I., & Raman, V. (2008). Measuring author contributions to the Wikipedia. In *Proceedings of the 4th International Symposium on Wikis* (pp. 15:1–15:10). New York, NY, USA: ACM. doi:10.1145/1822258.1822279
- Aha, D. W., & Kibler, D. (1991). Instance-based learning algorithms. In *Machine Learning* (pp. 37–66).
- Anderson, M. L., Fister, A., Lee, B., & Wang, D. (2004). On the frequency and types of meta-language in conversation: A preliminary report. In *14th Annual Conference of the Society for Text and Discourse*.
- Anderson, M. L., Okamoto, Y. A., Josyula, D., & Perlis, D. (2002). The Use-Mention Distinction and Its Importance to HCI. In *Proceedings of the Sixth Workshop on the Semantics and Pragmatics of Dialog*, 21–28.
- Audi, R. (1995). *The Cambridge Dictionary of Philosophy*. Cambridge University Press.
- Bird, S. (2006). NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions* (pp. 69–72). Stroudsburg, PA, USA: Association for Computational Linguistics.
- Cappelen, H., & Lepore, E. (1997). Varieties of quotation. *Mind*, 106(423), 429–450. doi:10.1093/mind/106.423.429
- Davidson, D. (1979). Quotation. *Theory and Decision*, 11(1), 27–40. doi:10.1007/BF00126690
- Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge: MIT Press.
- García-Carpintero, M. (2004). The deferred ostension theory of quotation. *Noûs*, 38(4), 674–692.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11, 10–18. doi:10.1145/1656274.1656278
- Hu, G. (2010). A place for metalanguage in the L2 classroom. *ELT Journal*. doi:10.1093/elt/ccq037
- Jaworski, A., Coupland, N., & Galasinski, D. (Eds.). (2004). *Metalanguage: Language, Power, and Social Process*. De Gruyter.
- John, G., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338–345). Morgan Kaufmann.
- Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation*, 13, 637–649. doi:10.1162/089976601300014493
- Kohavi, R. (1995). The Power of Decision Tables. In *Proceedings of the European Conference on Machine Learning* (pp. 174–189). Springer Verlag.
- Levy, R., & Andrew, G. (2006). TRegex and TSurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.
- Maier, E. (2007). Mixed quotation: between use and mention. In *Logic and Engineering of Natural Language Semantics Workshop*. Retrieved from [http://ncs.ruhosting.nl/emar/em\\_lens\\_quot.pdf](http://ncs.ruhosting.nl/emar/em_lens_quot.pdf)
- Marneffe, M. D., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 449–454.
- Punyakanok, V., Roth, D., & Yih, W. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2), 257–287. doi:10.1162/coli.2008.34.2.257
- Quine, W. V. O. (1940). *Mathematical logic*. Cambridge, MA: Harvard University Press.
- Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. San Mateo, Calif.: Morgan Kaufmann Publishers.
- Saka, P. (2006). The Demonstrative and Identity Theories of Quotation. *Journal of Philosophy*, 103(9), 452–471.
- Sperber, D., & Wilson, D. (1981). Irony and the Use-Mention Distinction. In *Radical Pragmatics* (pp. 295–318). New York.
- Tarski, A. (1933). The concept of truth in formalized languages. In J. H. Woodger (Ed.), *Logic, Semantics, Mathematics*. Oxford: Oxford University Press.
- Wilson, S. (2011a). *A computational theory of the use-mention distinction in natural language*. University of Maryland at College Park. PhD Thesis, College Park, MD, USA.
- Wilson, S. (2011b). In search of the use-mention distinction and its impact on language processing tasks. *International Journal of Computational Linguistics and Applications*, 2(1-2), 139–154.
- Wilson, S. (2012). The Creation of a Corpus of English Metalanguage. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, (pp. 638–646).

# On the Effectiveness of Using Syntactic and Shallow Semantic Tree Kernels for Automatic Assessment of Essays

**Yllias Chali**

University of Lethbridge  
Lethbridge, AB, Canada  
chali@cs.uleth.ca

**Sadid A. Hasan**

University of Lethbridge  
Lethbridge, AB, Canada  
hasan@cs.uleth.ca

## Abstract

This paper is concerned with the problem of automatic essay grading, where the task is to grade student written essays given course materials and a set of human-graded essays as training data. Latent Semantic Analysis (LSA) has been used extensively over the years to accomplish this task. However, the major limitation of LSA is that it only retains the frequency of words by disregarding the word sequence, and the syntactic and semantic structure of texts. As a remedy, we propose the use of syntactic and shallow semantic tree kernels for grading essays. Experiments suggest that syntactic and semantic structural information can significantly improve the performance of the state-of-the-art LSA-based models for automatic essay grading.

## 1 Introduction and Related Work

To evaluate the content of free texts is a challenging task for humans. Automation of this process is useful when an expert evaluator is unavailable in today's Internet-based learning environment. Research to automate the assessment of free texts, such as grading student-written essays, has been carried out over the years (Kakkonen et al., 2006; Kakkonen and Sutinen, 2004; Kanejiya et al., 2003; Persing et al., 2010; Yannakoudakis et al., 2011). Some notable essay scoring systems currently available are *AutoScore* by American Institutes for Research (AIR), *Bookette* by CTB/McGraw-Hill, *Project Essay Grade* by Measurement, Inc. and *Intelligent Essay Assessor* by Pearson Knowledge Technologies. The approaches such as Project Essay Grade and e-rater were solely based on some simple surface features that took essay-length, number of commas etc. into consideration (Page and Petersen,

1995; Powers et al., 2000). The major drawback of these systems is that they ignore the creativity factor by only dealing with the simple measures. To overcome this limitation, recent researches tend to focus on understanding the inner meaning of the texts. Latent Semantic Analysis (LSA) (Landauer et al., 1998; Deerwester et al., 1990) has been shown to fit well in addressing this task (Kakkonen et al., 2006; Kakkonen and Sutinen, 2004; Lintean et al., 2010; Kanejiya et al., 2003).

LSA uses a sophisticated approach to decode the inherent relationships between a context (typically a sentence, a paragraph or a document) and the words that they contain. This approach is based on Bag-Of-Words (BOW) assumption that uses the frequency of occurrence of each word in the context to construct a word-by-context co-occurrence matrix (Kanejiya et al., 2003). The major limitation of LSA is that it only retains the frequency of the words and does not take into account the sequence of them (word ordering). It ignores the syntactic and semantic structure of the context and thus, cannot distinguish between "The police shot the gunman" and "The gunman shot the police". Traditionally, information extraction techniques are based on the BOW approach augmented by language modeling. But when the task like *automated essay grading* requires the evaluation of more complex syntactic and semantic structures, the approaches based on only BOW are often inadequate to perform fine-level textual analysis. For example, in the basic LSA model for automated essay grading, a student essay can obtain a good grade by having a very small number of highly representative words that correlates the golden essays. This also means that the repetition of important terms without having any syntactic/semantic appropriateness can lead to an overstated grade (Jorge-Botana et al., 2010).

Several improvements on BOW have been shown by the use of dependency trees and syntac-

tic parse trees over the years (Hirao et al., 2004; Punyakanok et al., 2004; Kim and Kim, 2010). Kakkonen et al. (2006) used an enhanced LSA approach by incorporating parts-of-speech (POS) information to improve the performance of the basic LSA model for automatic essay grading. The augmentation of POS information into the basic LSA model enabled it to exploit a sufficient amount of local information about internal relations among the words. In this manner, the enhanced LSA model could disambiguate the meaning between the words having the same base forms but different POS tags. Kanejiya et al. (2003) proposed a similar model called *Syntactically Enhanced LSA* by considering a word along with its syntactic neighborhood (obtained from the part-of-speech tag of its preceding word). Wiemer-Hastings and Zipitria (2001) showed that a sentence comparison metric that combines structure-derived information with vector-based semantics has a better correlation to human judgements than the LSA model alone. This motivates us to propose the use of syntactic and semantic structural information (by means of syntactic and shallow semantic tree kernels) with a LSA-based model to automatically grade essays. The effectiveness of using various text-to-text semantic similarity measures, and dependency graph alignment techniques have been also shown to improve upon the BOW approaches for a similar task of short answer grading (Mohler et al., 2011; Mohler and Mihalcea, 2009).

The importance of syntactic and semantic features in finding textual similarity is described by Moschitti et al. (2007), and Moschitti and Basili (2006). An effective way to integrate syntactic and semantic structures in different applications is the use of *tree kernel* functions (Collins and Duffy, 2001), which has been successfully applied to other Natural Language Processing (NLP) tasks such as question classification (Moschitti and Basili, 2006). In this paper, we use the tree kernel functions and to the best of our knowledge, no other study has used tree kernel functions before to encode syntactic/semantic information for more complex tasks such as computing the relatedness between the contexts for automatic essay grading. Our experiments on an occupational therapy dataset show that the addition of syntactic and semantic information can improve the performance of the BOW-based and POS enhanced state-of-the-art LSA models significantly.

## 2 LSA Model for Essay Grading

LSA can determine the similarity of the meaning of words and the context based on word co-occurrence information (Kakkonen et al., 2006). Our grading model is most closely related to the approach described in Kakkonen and Sutinen (2004) where the experiments were conducted in the Finnish language. However, in this work, we experiment with the essays and course materials written in the English language. The main idea is based on the assumption that a student's knowledge is largely dependent on learning the course content; therefore, the student's knowledge can be computed as the degree of semantic similarity between the essay and the given course materials. An essay will get a higher grade if it closely matches with the course content.

The grading process includes three major steps. In the first step, we build a semantic space from the given course materials by constructing a word-by-context matrix (WCM). Here we use different local and global weighting functions to build several LSA models (for baseline selection). In the next step, a set of pre-scored (human-graded) essays are transformed into a query-vector form similar to each vector in the WCM and then their similarity with the semantic space is computed in order to define the threshold values for each grade category. The similarity score for each essay is calculated by using the traditional cosine similarity measure. In the last step, the student-written to-be-graded essays are transformed into the query-vector forms and compared to the semantic space in a similar way. The threshold values for the grade categories are examined to specify which essay belongs to which grade category.

As discussed previously, the basic LSA model for automatic essay grading lacks sensitivity to the context in which the words appear since it is solely based on the BOW assumption. It ignores the internal structure of the sentences and does not consider word orders. Our aim in this paper is to propose a similarity measure in which syntactic and/or semantic information can be added to enhance the basic LSA model by encoding the relational information between the words in sentences. We claim that for a complex task like evaluating student-written essays, where the relatedness between the sentences of an essay and the given course materials is an important factor, our grading model would perform more effectively if

we could incorporate the syntactic and semantic information with the standard cosine measure (i.e. done in basic LSA) while calculating the similarity between sentences. In the next sections, we describe how we can encode syntactic and semantic structures in calculating the similarity between sentences.

### 3 Syntactic Similarity Measure (SYN)

Inspired by the potential significance of using syntactic measures for finding similar texts, we get a strong motivation to use it as a similarity measure in essay grading framework. The first step to calculate the syntactic similarity between two sentences is to parse the corresponding sentences into syntactic trees using the Charniak parser (Charniak, 1999). Once we build the syntactic trees, our next task is to measure the similarity between the trees. For this, every tree  $T$  is represented by an  $m$  dimensional vector  $v(T) = (v_1(T), v_2(T), \dots, v_m(T))$ , where the  $i$ -th element  $v_i(T)$  is the number of occurrences of the  $i$ -th tree fragment in tree  $T$  (Moschitti et al., 2007). The tree kernel of two trees  $T_1$  and  $T_2$  is actually the inner product of  $v(T_1)$  and  $v(T_2)$  (Collins and Duffy, 2001), which computes the number of common subtrees between two trees to provide the similarity score between a pair of sentences. Each course material sentence contributes a score to the essay sentences. The average syntactic similarity scores of the essay sentences are combined to get an overall similarity score for an essay with respect to the course material sentences.

### 4 Semantic Similarity Measure (SEM)

Shallow semantic representations can prevent the weakness of cosine similarity based models (Moschitti et al., 2007). Since the textual similarity between a pair of sentences relies on a deep understanding of the semantics of both, applying semantic similarity measurement in our essay grading framework is another noticeable contribution of this paper. To calculate the semantic similarity between two sentences, we first parse the corresponding sentences semantically using the Semantic Role Labeling (SRL) system, ASSERT<sup>1</sup>. We represent the annotated sentences using tree structures called semantic trees (ST). In the tree kernel method (Section 3), common substructures cannot

<sup>1</sup>Available at <http://cemantix.org/assert>

be composed of a node with only some of its children. Moschitti et al. (2007) solved this problem by designing the Shallow Semantic Tree Kernel (SSTK) which allows to match portions of a ST. The SSTK function yields the similarity score between a pair of sentences based on their semantic structures. An overall semantic similarity score for each essay is obtained similarly as the syntactic measure.

## 5 Experiments and Evaluation

### 5.1 Data

We use a dataset obtained from an occupational therapy course where 3 journal articles are provided as the course materials. The students are asked to answer an essay-type question. The dataset contains 91 student-written essays, which are graded by a professor<sup>2</sup>. The length of the essays varied from 180 to 775 characters. We use 3-fold cross-validation for our experiments.

### 5.2 System Settings

Initially, we split the course materials into 64 paragraphs and built the word-by-paragraph matrix by treating the paragraphs as contexts. Our preliminary experiments suggested that this scheme shows worse performance than that of using individual sentences as the contexts. So, we tokenized the course materials (journal articles) into 741 sentences and built the word-by-sentence matrix. We do not perform word stemming for our experiments. We use a stop word list of 429 words to remove any occurrence of them from the datasets. In this work, C++ and Perl are used as the programming languages to implement the LSA models and encode the syntactic and shallow semantic structures. The GNU Scientific Library (GSL<sup>3</sup>) software package is used to perform the SVD calculations in LSA. During the dimensionality reduction step of LSA, we have experimented with different dimensions of the semantic space. Finally, we kept 100 as the number of dimensions since we got better results using this value. We experiment with six variations of the LSA model based on different local and global weighting functions according to Chali and Hasan (2012). The best performing LSA model is used as the baseline for comparison purposes.

<sup>2</sup>Each essay is graded on a scale from 0 to 6.

<sup>3</sup><http://www.gnu.org/software/gsl/>

### 5.2.1 Variations of the LSA Model

Inspired by the work of Jorge-Botana et al. (2010), we experiment with different local and global weighting functions applied to the WCM. The main idea is to transform the raw frequency cell  $x_{ij}$  of the WCM into the product of a local term weight  $l_{ij}$ , and a global term weight  $g_j$ . Given the term/document frequency matrix (WCM), a weighting algorithm is applied to each entry that has three components to make up the new weighted value in the term/document matrix. This looks as:  $w_{ij} = l_{ij} * g_j * N_j$ , where  $w_{ij}$  is the weighted value for the  $i^{th}$  term in the  $j^{th}$  context,  $l_{ij}$  is the local weight for term  $i$  in the context  $j$ ,  $g_j$  is the global weight for the term  $i$  across all contexts in the collection, and  $N_j$  is the normalization factor for context  $j$ .

**Local Weighting:** We use two local weighting methods in this work: 1) *Logarithmic*:  $\log(1 + f_{ij})$ , and 2) *Term Frequency (TF)*:  $f_{ij}$ , where  $f_{ij}$  is the number of times (frequency) the term  $i$  appears in the context  $j$ .

**Global Weighting:** We experiment with three global weighting methods: 1) *Entropy*:  $1 + \left(\frac{\sum_j (p_{ij} \log(p_{ij}))}{\log(n)}\right)$ , 2) *Inverse Document Frequency (IDF)*:  $\log\left(\frac{n}{df_i}\right) + 1$ , and 3) *Global Frequency/Inverse Document Frequency (GF/IDF)*:  $\frac{\sum_j f_{ij}}{df_i}$ , where  $p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}}$ ,  $n$  is the number of documents in our word by context matrix, and  $df_i$  is the number of contexts in which the term  $i$  is present.

**Different Models:** By combining the different local and global weighting schemes, we build the following six different LSA models: **1) LE:** logarithmic local weighting and entropy-based global weighting, **2) LI:** logarithmic local weighting and IDF-based global weighting, **3) LG:** logarithmic local weighting and GF/IDF-based global weighting, **4) TE:** TF-based local weighting and entropy-based global weighting, **5) TI:** TF-based local weighting and IDF-based global weighting, and **6) TG:** TF-based local weighting and GF/IDF-based global weighting.

### 5.2.2 Systems for Evaluation

To study the impact of syntactic and semantic representation introduced earlier (in Section 3 and Section 4) for the essay grading task, we build six systems as defined below:

**(1) Baseline:** Our baseline is the best performing

LSA model among the six variations (discussed in Section 5.2.1) that uses the standard cosine similarity measure based on BOW assumption and does not consider syntactic/semantic information.

**(2) SYN:** This system measures the similarity between the sentences using the *syntactic tree* and the *general tree kernel* function defined in Section 3.

**(3) SEM:** This system measures the similarity between the sentences using the *shallow semantic tree* and the *shallow semantic tree kernel* function defined in Section 4.

**(4) LSA+SYN:** This system measures the similarity between the sentences using both standard cosine similarity measure and the syntactic tree kernel.

**(5) LSA+SEM:** This system measures the similarity between the sentences using both standard cosine similarity measure and the shallow semantic tree kernel.

**(6) LSA+SYN+SEM:** This system measures the similarity between the sentences using standard cosine similarity measure, syntactic tree kernel, and shallow semantic tree kernel.

We use an equally weighted linear combination by summing the similarity scores obtained by **LSA**, **SYN** and **SEM** (when multiple similarity measures are used) as we believe that the word distribution, syntactic and semantic similarity between a pair of texts are all equally important. The average value of the similarity scores of the representative essays (with comparison to the course materials) of a certain grade category is considered as the threshold for that particular grade. For example, if we have five pre-scored essays of grade 6, we obtain five similarity scores corresponding to the course materials. The average of these scores are considered as the minimum score (threshold) that should be obtained by a non-graded student-written essay in order to assign it the grade 6. For a more robust evaluation, we also implement a state-of-the-art part-of-speech (POS) enhanced LSA model (**POS+LSA**) for essay grading according to Kakkonen et al. (2006) by considering the POS tag of the current word.

### 5.3 Evaluation Results

In Table 1, we present the results of our baseline selection step. The first column stands for the weighting model used ("N" denotes no weighting method applied). The "Correlation" column

presents the Spearman rank correlation between the scores given by the professor and the systems. The “Accuracy” column stands for the proportion of the cases where the professor and the system have assigned the same grade whereas the next column shows the percentage of essays where the system-assigned grade is at most one point away or exactly the same as the professor. From these results, we can see that the performance of the systems varied (having correlation from 0.32 to 0.68) with respect to the weighting scheme applied. We observe that the combination of the logarithmic local weighting with the entropy-based global weighting scheme performs the best for our dataset. Hence, we use this model as our baseline system.

In Table 2, we present the results of different systems. The columns denote the same meaning as Table 1. We can see that for the **SYN** system, the correlation is decreased by 7.93% from the baseline and 12.69% from the **POS+LSA** system. The **SEM** system improves the correlation over the baseline system by 2.94%, but decreases by 1.42% from the **POS+LSA** system. The **LSA+SYN** system improves the correlation over the baseline system by 7.35% and over the **POS+LSA** system by 2.81% whereas the **LSA+SEM** system improves the correlation by 11.76%, and 7.04% respectively. Lastly, the **LSA+SYN+SEM** system improves the correlation over the baseline system by 10.29% and over the **POS+LSA** system by 5.63%. Analysis of these results reveals that the proposed systems (that encode the syntactic and/or semantic information with the basic LSA model) considerably outperform both the standard cosine similarity based and the state-of-the-art POS enhanced LSA approaches. The results also denote that encoding the syntactic and/or semantic information on top of the standard cosine similarity measure often outperform the systems that consider only syntactic and/or semantic information.

**Statistical Significance:** We use Student’s t-test to compute whether the differences between the correlations of different systems are statistically significant. For this computation, we have one measurement variable, “correlation”, and one nominal variable, “system”. We had three runs and the observations were the set of correlations for each of the systems in consideration. We find that the differences between the correla-

tions are statistically significant at  $p < 0.05$  except for the differences between the **SEM** system and the **POS+LSA** system, and between the **LSA+SYN+SEM** system and the **LSA+SEM** system. We also compute the statistical significance of the correlations themselves. In Table 1, the reported correlations are statistically significant ( $p < 0.05$ ) except for “TE” and “N” models. The correlations reported in Table 2 are statistically significant ( $p < 0.05$ ).

Model	Corr.	Accuracy (%)	Close (%)
LE	0.68	40.2	73.1
LI	0.49	27.1	51.8
LG	0.40	21.3	42.2
TE	0.34	19.2	36.4
TI	0.52	32.6	58.6
TG	0.38	20.4	38.9
N	0.32	17.8	32.9

Table 1: Variations of LSA model

System	Corr.	Accuracy (%)	Close (%)
Baseline	0.68	40.2	73.1
POS+LSA	0.71	42.6	70.8
SYN	0.63	34.8	60.1
SEM	0.70	41.5	76.2
LSA+SYN	0.73	43.2	78.1
LSA+SEM	0.76	48.3	82.5
LSA+SYN+SEM	0.75	46.7	79.6

Table 2: Evaluation results

## 5.4 Discussion

### 5.4.1 Is Thresholding Adequate?

Our experiments showed that the formation of the thresholds were adequate as we could obtain different thresholds for different grade categories. However, in a few cases, the difference between two subsequent thresholds was found to be small. This might be because the grades were not evenly distributed among the given human-graded corpus. Ideally it is desirable to have the representative training essays across the spectrum of possible grades to set the thresholds on by using the SVD generated from the training materials. We also believe that the use of a larger dataset while defining the thresholds might improve the overall performance. Our further experiments (shown in the next subsection) support this claim. The length of the essays is another issue since longer essays tend to capture more information in their representative vectors which provides the scope for a better similarity matching with the semantic space.



### 5.4.2 Can We Automate Data Generation?

To experiment with an LSA-based model we require a number of student-written essays. It is often hard to collect a huge number of raw student-written essays and process them into the machine-readable format. To reduce the human intervention involved in producing a large amount of training data, we propose to automate this process by using the ROUGE (Lin, 2004) toolkit. We assume each individual sentence of the course material as the candidate extract sentence and calculate its ROUGE similarity scores with the corresponding golden essay. Thus an average ROUGE score is assigned to each sentence of the course content. We choose the top 50% sentences based on ROUGE scores to have the label +1 (candidate essay sentences) and the rest to have the label -1 (non-essay sentences), and thus, we generate essays up to a predefined word limit considering different levels of expertise of the students. The sentences having the label +1 are further sorted in descending order of their assigned scores. A collection of sentences (upto length 775 characters) having the highest scores are considered to have the grade 6, the next collection of sentences to grade 5 and so on. In this manner, we have generated 216 essays from the given course materials. We have used 20 golden essays in this experiment. We treated the essays that got the full score of 6 as the golden essays. The automatically generated essays appeared to be similar in content to that of the original student-written essays.

We run further experiments using the automatically generated dataset in order to make sure that the proposed methods are useful for the essay grading task. For this purpose, we build a corpus containing 147 essays (that include both human-written and automatic essays), where the grade categories are evenly distributed. We use 3-fold cross-validation for our experiments. In Table 3, we present the results of different systems. A relative comparison of these results with the results of Table 2 yields that there is a marginal improvement in the overall performance of all the systems except for the **LSA+SYN** system. This phenomenon suggests that the even distribution of the grade categories in a larger corpus of essays is useful in general to achieve better grading performance. The results also reveal the effectiveness of our proposed method for automatic training data generation. The differences between

the correlations are statistically significant at  $p < 0.05$  (using Student's t-test) except for the differences between the **LSA+SYN** system and the baseline, and between the **LSA+SYN+SEM** system and the **LSA+SEM** system. The reported correlations are also found to be statistically significant ( $p < 0.05$ ).

System	Corr.	Accuracy (%)	Close (%)
Baseline	0.71	42.6	75.4
POS+LSA	0.73	45.2	72.5
SYN	0.65	35.2	63.5
SEM	0.75	48.5	79.7
LSA+SYN	0.72	42.8	77.5
LSA+SEM	0.80	52.3	84.2
LSA+SYN+SEM	0.78	50.1	81.6

Table 3: Evaluation results (second corpus)

## 6 Conclusion and Future Work

We proposed to encode the syntactic and semantic information for measuring sentence relationships to automatically grade student-written essays and demonstrated that adding syntactic and/or semantic information on top of the standard cosine measure improves the performance over the BOW based and state-of-the-art POS enhanced LSA models. To the best of our knowledge, no other study has used syntactic and shallow semantic tree kernels for the task of automatic essay grading to improve the basic LSA model's performance. Our approach to automate the data generation process is also unique and novel in this problem domain. Experimental results revealed the effectiveness of the proposed approach. Our experiments also suggested that the overall syntactic/semantic similarity between a pair of texts can be effectively captured using the aggregated tree kernel scores of all possible sentence pairs. In the future, we plan to focus on other important metrics in terms of creativity, novelty, etc. for the essay grading task which we believe would further enhance the overall grading performance given that the major limitation of the basic LSA model is overcome.

### Acknowledgments

The research reported in this paper was supported by the Mitacs-Accelerate internship program, the Natural Sciences and Engineering Research Council (NSERC) of Canada – discovery grant and the University of Lethbridge. The authors are grateful to Colin Layfield and Laurence Meadows for their assistance.

## References

- Y. Chali and S. A. Hasan. 2012. Automatically Assessing Free Texts. In *Proceedings of the Workshop on Speech and Language Processing Tools in Education*, pages 9–16, Mumbai, India. COLING 2012.
- E. Charniak. 1999. A Maximum-Entropy-Inspired Parser. In *Technical Report CS-99-12*, Brown University, Computer Science Department.
- M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.
- S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- T. Hirao, J. Suzuki, H. Isozaki, and E. Maeda. 2004. Dependency-based Sentence Alignment for Multiple Document Summarization. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 446–452, Geneva, Switzerland.
- G. Jorge-Botana, J. A. Leon, R. Olmos, and I. Escudero. 2010. Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora. *Journal of Quantitative Linguistics*, 17(1):1–29.
- T. Kakkonen and E. Sutinen. 2004. Automatic Assessment of the Content of Essays Based on Course Materials. In *Proceedings of the 2nd IEEE International Conference on Information Technology: Research and Education*, pages 126–130.
- T. Kakkonen, N. Myller, and E. Sutinen. 2006. Applying Part-Of-Speech Enhanced LSA to Automatic Essay Grading. In *Proceedings of the 4th IEEE International Conference on Information Technology: Research and Education (ITRE 2006)*.
- D. Kanjija, A. Kumar, and S. Prasad. 2003. Automatic Evaluation of Students' Answers using Syntactically Enhanced LSA. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing - Volume 2*, pages 53–60. ACL.
- Y. Kim and Y. Kim. 2010. An Autonomous Assessment System based on Combined Latent Semantic Kernels. *Expert Systems with Applications*, 37(4):3219–3228.
- T. Landauer, P. Foltz, and D. Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2):259–284.
- C. Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain.
- M. C. Lintean, C. Moldovan, V. Rus, and D. S. McNamara. 2010. The Role of Local and Global Weighting in Assessing the Semantic Similarity of Texts Using Latent Semantic Analysis. In *FLAIRS Conference*.
- M. Mohler and R. Mihalcea. 2009. Text-to-Text Semantic Similarity for Automatic Short Answer Grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 567–575. ACL.
- M. Mohler, R. Bunescu, and R. Mihalcea. 2011. Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 752–762. ACL.
- A. Moschitti and R. Basili. 2006. A Tree Kernel Approach to Question and Answer Classification in Question Answering Systems. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy.
- A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 776–783, Prague, Czech Republic.
- E. B. Page and N. S. Petersen. 1995. The Computer Moves into Essay Grading: Updating the Ancient Test. *Phi Delta Kappan*, 76(7).
- I. Persing, A. Davis, and V. Ng. 2010. Modeling Organization in Student Essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239. ACL.
- D. E. Powers, J. Burstein, M. Chodorow, M. E. Fowles, and K. Kukich. 2000. Comparing the Validity of Automated and Human Essay Scoring. (*GRE No. 98-08a, ETS RR-00-10*). Princeton, NJ: Educational Testing Service.
- V. Punyakanok, D. Roth, and W. Yih. 2004. Mapping Dependencies Trees: An Application to Question Answering. In *Proceedings of AI & Math*, Florida, USA.
- P. Wiemer-Hastings and I. Zipitria. 2001. Rules for Syntax, Vectors for Semantics. In *Proceedings of the Twenty-third Annual Conference of the Cognitive Science Society*, pages 1112–1117. Erlbaum.
- H. Yannakoudakis, T. Briscoe, and B. Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th ACL-HLT*, pages 180–189. ACL.

# Little by Little: Semi Supervised Stemming through Stem Set Minimization

Vasudevan N, Pushpak Bhattacharyya

Dept. of Computer Science and Engg.

IIT Bombay, Mumbai

{vasudevan,pb}@cse.iitb.ac.in

## Abstract

In this paper we take an important step towards completely unsupervised stemming by giving a scheme for semi supervised stemming. The input to the system is a list of word forms and suffixes. The motivation of the work comes from the need to create a root or stem identifier for a language that has electronic corpora and some elementary linguistic work in the form of, say, suffix list. The scope of our work is suffix based morphology, (*i.e.*, no prefix or infix morphology). We give two greedy algorithms for stemming. We have performed extensive experimentation with four languages: English, Hindi, Malayalam and Marathi. Accuracy figures ranges from 80% to 88% are reported for all languages.

## 1 Introduction

Stemming is critical for many NLP, IR and IE problems (Hull, 1996). In the current paper, we report construction of a semi supervised stemmer that does stemming by *minimizing the total number of distinct stems*. The input to the system is the word list along with the legal suffix list of the language. Even if a language does not have an elaborate linguistic tradition and exhaustive body of linguistic work, the language is expected to have at least the legal suffix list for nouns and verbs.

To get the intuition behind our work, consider the word list {*boy, boys, moss, mosses*}. The splitting (the *split* is formally defined later) that generate the minimum number of stems (*viz.*, 2) from the above word list is {*boy+ $\phi$ , boy+s, moss+ $\phi$ , moss+es* }, where  $\phi$  is the null suffix. The minimum stem set is {*boy, moss*}. Any other splitting, say *mosse+s* will increase the number of stems.

This work is applicable to languages with concatenative morphology where suffixes stack one after another. However, problems arise when there are phonemic changes in the boundaries of the stems and suffixes (*sandhi*). In such situation, existence of fused, composite suffixes in the suffix list is assumed.

The roadmap of the paper is as follows. Related work in morphology learning is explained in section 2. Notations and terminologies used in this paper are defined in section 3. In section 4 we defined the stemming problem addressed in this work. Two models for this stemming are proposed in section 5 and section 6. In section 7, we described various experiments conducted. The conclusions and future works are presented in section 8.

## 2 Related Work

Morphology learning is one of the widely attempted problems in NLP. A recent survey by Harald Hammarström (2011) gives an overall view of unsupervised morphology learning. The Linguistica (Goldsmith, 2001) model based on minimum description length (MDL) principle is one of the benchmark works of unsupervised stemming. In the Linguistica model, the authors defined a signature structure. The objective of their stemming approach is the minimization of total description length, *i.e.*, the description length of stem list, suffix list, signatures and corpus.

Maximum a posteriori model (Creutz and Lagus, 2007) is a generalization of the Linguistica model, in the sense of being a recursive MDL. This probabilistic approach is more suitable for languages with more than one suffix. Stochastic transducer based model (Clark, 2001) and generative probabilistic model (Snover et al., 2002) are other relevant probabilistic models for stemming.

A Markov Random Field by Dreyer (2009) is also a useful probabilistic approach related to unsupervised morphology.

Graph based model (Johnson and Martin, 2003), lazy learning based model (van den Bosch and Daelemans, 1999), clustering based same stem identification model (Hammarström, 2006a; Hammarström, 2006b), ParaMor system for paradigm learning (Monson et al., 2008) and full morpheme segmentation and automatic induction of orthographic rules (Dasgupta and Ng, 2007; Dasgupta and Ng, 2006) are also a relevant.

### 3 Terminology and Notation

Let us define some terms and notations used throughout this paper.

$w$ : word

$W$ : input word list

$N$ : number of words in the input word list

$X$ : input suffix list

$x$ : suffix candidate of  $w$  (possible suffix)

$\phi$ : null suffix, *i.e.*, the suffix with zero length.

$t$ : stem candidate of  $w$  (possible stem)

$T$ : the set of possible stems from the word list

$|\cdot|$ : overloaded for the cardinality of a set and the length of a string

‘+’: splitting (breakage) of string

‘.’: concatenation of strings

*Stem*: the longest common prefix of the inflected words of a lexeme. The stem from the set of inflections of the lexeme *play* ( $\{plays, playing, played\}$ ) is *play*. Note that while a lexeme has to be a meaningful word in the language, the stem need not to be so. *E.g.*, stem of the words *lady* and *ladies* is *lad*, but the lexeme is *lady*.

*Suffix*: the portion(s) of the word after removing its stem. *E.g.*, the suffix of *boys* is *s*. The suffix can be a null string ( $\phi$ ) or chain of suffixes (Words in agglutinative languages can have multiple suffixes. In this case, chain of suffixes is taken as a single suffix.).

*Split*: the outcome of the process of segmentation (null strings permitted). A word can be segmented in multiple ways, giving rise to multiple *splits*. *E.g.*, a *split* of *boys* can be *b+oys*, *bo+ys*, *boy+s* or *boys+φ*. The *correct split* is the *split* that

separates a word in to its correct stem and correct suffix. *E.g.*, the *boy+s* is the *correct split* of *boys*.

*Splitset*: a set of *splits* obtained from the whole input word list. For every word in the word list, exactly one *split* will be there in the *splitset*. In other words,  $splitset = \{t + x \mid t \cdot x \in W \text{ and for any } t' + x' \in splitset, t \cdot x = t' \cdot x' \rightarrow t = t' \text{ and } x = x'\}$ . *E.g.*,  $\{bo+ys, girl+\phi, play+ing\}$  is a *splitset* of  $\{boys, girl, playing\}$ . The *correct splitset* is defined as the set of *correct splits* of all the words from the given word list. The *correct splitset* of  $\{boys, girl, playing\}$  is  $\{boy+s, girl+\phi, play+ing\}$ .

$T_s(splitset)$ : the set of stems from the *splitset*. *E.g.*,  $T_s(\{bo+ys, girl+\phi, play+ing\}) = \{bo, girl, play\}$ .

$X_s(splitset)$ : the set of suffixes from the *splitset*. *E.g.*,  $X_s(\{bo+ys, girl+\phi, play+ing\}) = \{ys, \phi, ing\}$ .

### 4 Problem Definition

The stemming problem addressed in this paper is defined as follows. Given a list of word forms  $W$  and a list of suffixes  $X$ , the problem is to find the *correct splitset*.

The suffix list of a language plays a crucial role in this problem. The suffix list considered in this problem should contain all atomic suffixes (*E.g.*, *s, es, ing*) and its orthographic variants (*E.g.*, *iness* in *happiness*). The suffix list also should contain chain of suffixes in case of agglutination. *E.g.*, the concatenated form of Malayalam<sup>1</sup> plural marker  $\text{കാല}(kal)$  and genitive case marker  $\text{ഉടെ}(ude)$  is  $\text{കാലുടെ}(kalude)$ . This concatenated form should be in the suffix list since it is the suffix (as per our definition) of the word  $\text{കുട്ടികളുടെ}(kuttikalude)$ (of children).

The suffix list  $X$  should be as large as possible and  $X$  should be a superset of all suffixes in the word list, *i.e.*,  $X_s(correct\ splitset) \subseteq X$ . *E.g.*, for the sample word list  $W = \{boy, boys, moss, mosses\}$ , the set  $X_s(correct\ splitset)$  is  $\{\phi, s, es\}$ , where  $\phi$  is the null suffix. So the suffix list should contain at least  $\phi, s$  and *es*.

The desired output of the above input is its *correct splitset*, *i.e.*,  $\{boy+\phi, boy+s, moss+\phi,$

<sup>1</sup>A morphologically rich language of India belonging to the Dravidian family.

*moss+es*}. Two computational models for this stemming problem is proposed in the section 5 and section 6.

## 5 Minimum Stem Set Model for Stemming

Consider the sample word list  $\{boy, boys, moss, mosses\}$  and suffix list  $\{\phi, s, es\}$ . Out of all possible *splitsets* of this input, the *correct splitset*,  $\{boy+\phi, boy+s, moss+\phi, moss+es\}$  produces the minimum number of distinct stems. This intuition leads to the Minimum Stem Set model for stemming. The Minimum Stem Set model (MSS) identifies the *correct splitset* by minimizing the number of distinct stems. In other words, this model identifies the *splitset* with the minimum number of distinct stems as the *correct split*.

Core of the MSS model is an optimization problem (MSS problem). The MSS problem is formally stated as follows,

*Input:* A list of word forms ( $W$ ) and a set of suffixes ( $X$ ) such that  $X_s(\text{correct splitset}) \subseteq X$

*Output:* 
$$\underset{\text{splitset}: X_s(\text{splitset}) \subseteq X}{\text{argmin}} \left\{ |T_s(\text{splitset})| \right\}$$

### 5.1 Greedy Algorithm for MSS

Since the complexity of computing the MSS problem is NP-Hard (Vasudevan and Bhattacharyya, 2012), we designed an approximation algorithm by utilizing similarity between our problem and the set cover problem. Set cover problem is a well known NP-Hard problem, which has a simple greedy approximation algorithm with approximation factor of  $\log(N)$  (Chvatal, 1979). This approximation factor is the best attainable factor for the set cover problem (Feige, 1998). The corresponding greedy algorithm for MSS problem is the best polynomial time approximation algorithm. This greedy approximation algorithm for the MSS problem (Approx-MSS) is described below.

Input of the Approx-MSS algorithm is a word list  $W$  and a suffix list  $X$ . The algorithm first initializes a set of all possible stems  $T$ . This can be done by stripping suffixes in  $X$  from end of each word in  $W$ . Then it initialize sets of all possible inflections of each  $t$  in  $T$ , let's call  $Infl(t)$ .  $Infl(t)$  can be initialized by appending suffixes from  $X$  to  $t$ . If a word created by appending a suffix  $x$  in  $X$  to

$t$  is in  $W$ , then add that word into  $Infl(t)$ . Set of all possible stems ( $T$ ) and  $Infl(t)$  of the running example is shown in Table 1.

After the initialization, the algorithm start the iterations with an empty *splitset*. In the first step, it chooses a stem  $t$  from  $T$  that has maximum  $|Infl(t) \cap W|$ . I.e., it finds a  $t^* = \underset{t \in T}{\text{argmax}} \{|Infl(t) \cap W|\}$ . In the next step, for all words ( $w$ ) in  $Infl(t^*) \cap W$ , the *split*  $t^* + x$  is added to *splitset*, where  $x$  is the suffix of word  $w$  after the stem  $t^*$ . Then it removes all words from  $W$  whose *splits* are added to *splitset*. This process is repeated until the *splitset* is complete, i.e., for all words there is a *split* in the *splitset*. The complexity of this approximation algorithm is  $O(|W||X|)$  and approximation factor is  $\log(|W|)$ .

Consider the example shown in Table 1. Initially both *boy* and *moss* have highest  $|Infl(t) \cap W|$ . So the greedy algorithm chooses either one of them in the first step as  $t^*$ . In the next step it chooses the other one. By these two steps, the greedy algorithm identifies the *correct split* of all four words.

T	<i>mosses</i>	<i>mosse</i>	<i>moss</i>	<i>mos</i>	<i>boys</i>	<i>boy</i>
$Infl(t)$	{ <i>mosses</i> }	{ <i>mosses</i> }	{ <i>mosses</i> , <i>moss</i> }	{ <i>moss</i> }	{ <i>boys</i> }	{ <i>boys</i> , <i>boy</i> }

Table 1: Possible Stems, their  $Infl()$

## 6 Weighted Minimum Stem Set (WMSS) Model

MSS problem uses the information from other words to identify the stem of each word. If the word list doesn't have any other inflections of a word, then MSS cannot choose the stem properly. In this case MSS randomly selects one of the possible stems. This is one of the main drawbacks of MSS. Languages with poor morphology have a lesser number of inflections than that of language with rich morphology. So in a word list with fixed number of words, the above problem is more serious for morphologically poor languages.

We extended the MSS model to a Weighted Minimum Stem Set (WMSS) model, which reduces the number of distinct stems and the number of splits with non empty suffixes. Output of this model is also a *splitset*. Consider a small word list  $\{boy, boys, moss, mosses\}$  and a suffix list  $\{\phi, s, es, ses\}$ . In this case both  $\{boy+\phi, boy+s, moss+\phi,$

$moss+es$ } and  $\{boy+\phi, boy+s, mos+s, mos+ses\}$  are optimum solutions for MSS problem. In such a tie situation, the WMSS model prefer the *splitset* with more number of null suffix ( $\phi$ ), *i.e.*, the first one. From our knowledge about English language, we can see that the first one is the *correct splitset*.

In the WMSS model, a weight function  $wg(t)$  is defined for each and every possible stem  $t$  as  $wg(t) = 1 + \frac{[t \notin W]}{|W|}$ . Where  $[t \notin W]$  is the Iverson bracket (Weisstein, Online 30 04 2010), *i.e.*, it is 1 if  $t \notin W$ , 0 otherwise. WMSS will find out a *splitset* such that the total weight of all stems in  $T_s(splitset)$  is minimum. Let's define the problem in WMSS model formally.

*Input:* A list of word forms ( $W$ ) and a set of suffixes ( $X$ ) such that  $X_s(correct\ splitset) \subseteq X$

$$\text{Output: } \underset{splitset: X_s(splitset) \subseteq X}{\operatorname{argmin}} \left\{ \sum_{t \in T_s(splitset)} wg(t) \right\}$$

In this extended problem formulation,  $wg(t)$  contain two terms. The first term, the constant 1 is for reducing the number of distinct stems and the second term,  $\frac{[t \notin W]}{|W|}$  is for reducing the number of *splits* with non empty suffixes. If there is no second term then  $wg(t) = 1$  and it is exactly the same as MSS problem.

Since the maximum value of the second term in WMSS is  $\frac{1}{|W|}$  and maximum number of stems in any  $T_s(splitset)$  is less than  $|W|$ ,  $|T_s(splitset)| < \sum_{t \in T_s(splitset)} wg(t) < |T_s(splitset)| + 1$ . Therefore any solution of WMSS should be a solution of MSS, but the reverse is false. Relevance of this WMSS problem comes only if there are multiple solutions for MSS problem.

Since the solution of WMSS problem is a solution of MSS problem, the reduction from MSS to WMSS is trivial. Suppose WMSS have a polynomial time algorithm, then we can use that algorithm for MSS problem also. Since MSS is NP-Hard we can say that, WMSS is also NP-Hard.

### 6.1 Greedy Algorithm for WMSS

The WMSS problem can be solved effectively by utilizing its similarity with weighted set cover problem. Weighted set cover problem is also an NP-Hard problem, and its greedy approximation have a bound of  $\log(N)$ . The greedy algorithm for weighted set cover problem is adapted for WMSS

problem. The corresponding greedy algorithm for WMSS (Approx-WMSS) is explained below.

The Approx-WMSS is similar to Approx-MSS. The only difference is in the first step. While Approx-MSS algorithm selects a stem with maximum  $|Infl(t) \cap W|$  in the first step, Approx-WMSS algorithm selects a stem with maximum  $\frac{|Infl(t) \cap W|}{wg(t)}$ . Note that, when  $wg(t)$  is 1 then both terms are the same. All remaining steps are the same for both algorithms. Similar to Approx-MSS algorithm, the complexity of this approximation algorithm is  $O(|W||X|)$  and approximation factor is  $\log(|W|)$ .

Consider the  $W = \{boy, boys, moss, mosses\}$  and  $X = \{\phi, s, es, ses\}$ . The set of all possible stems and its corresponding  $Infl()$  and  $wg()$  are shown in Table 2. Initially *boy* has the highest  $\frac{|Infl(t) \cap W|}{wg(t)}$ . So this greedy algorithm chooses the stem *boy* and add *boy+ $\phi$*  and *boy+s* to *splitset* in the first step. In the next step it chooses *moss* and add *moss+es* and *moss+ $\phi$*  to *splitset*. By these two steps, this greedy algorithm terminate by identifying *correct splitset*.

T	<i>mosses</i>	<i>mosse</i>	<i>moss</i>	<i>mos</i>	<i>boys</i>	<i>boy</i>
$Infl(t)$	{ <i>mosses</i> }	{ <i>mosses</i> }	{ <i>mosses</i> , <i>moss</i> }	{ <i>moss</i> }	{ <i>boys</i> }	{ <i>boys</i> , <i>boy</i> }
$wg(t)$	1	$1 + \frac{1}{4}$	1	$1 + \frac{1}{4}$	1	1

Table 2: Possible Stems, their  $Infl()$  and  $wg()$

## 7 Experimentation

Two new stemming systems based on the greedy algorithms for MSS problem and WMSS problem are implemented. Performances of these systems are evaluated for four languages from Indo-European family and Dravidian family. The selected languages are English, Hindi, Marathi and Malayalam, in the increasing order of morphological complexity. First three languages are from Indo-European family while the fourth language, Malayalam is a highly agglutinative language from Dravidian family. These spectrum of languages from different families with different morphological richness is necessary for the evaluation of the suitability of proposed models.

Performance of proposed models are compared with different baselines. The first baseline is a random stem selection, which randomly selects a *split* for each word such that the suffix in this *split* is in the input suffix list. The length of the suffix (or

stem) is another information that can provide second and third baselines. The second one selects a *split* for each word form that has the smallest stem, albeit with the suffix in the input suffix list. Similarly the third one selects the *split* with the largest stem.

Linguistica is an MDL based system that identifies stem of each word in a word list without using any other input. One of the heuristics used in Linguistica model is modified to make the fourth baseline. In the Linguistica heuristics, a probability is assigned to every *split* for every word. Then iteratively it learns the best probability distribution by optimizing a figure of merit, which is a function of length and frequency of morphemes. Since there is no need to consider any *split* with a suffix which is not in the input suffix list, the sample space can be minimized. Probability distribution after this modification is learned using the same iterative procedure as in Linguistica. We implemented this modified Linguistica algorithm and considered it as fourth baseline.

## 7.1 Data Analysis

Word list of size 10,000 distinct words in Unicode format were selected for English, Hindi, Marathi and Malayalam. English words are taken from Brown and BNC corpora (Francis and Kucera, 1964; Edition, 2007). Selected Hindi words are from tourism and news corpus. The source of Marathi words for experimentation is the corpora from the Indian Language Corpora Initiative (ILCI) project, which is a Government of India effort (<http://www.tdil.mit.gov.in>). Malayalam words are obtained from IITMK<sup>2</sup> and from various blogs and newspapers. For each words the correct stem as per the definition, *i.e.*, the largest prefix of all inflected forms of the lexeme, is identified for the evaluation. Suffix lists are mainly created from the words in the word list. By adding available suffixes from web, the suffix lists are expanded as big as possible.

Counts and frequencies of stems and suffixes are relevant statistics to reflect the nature of word list for stemming. So the number of distinct stems (*StCount*) and suffixes (*SfCount*) are counted from each word list. The average stem frequencies (*StFreq*) and average suffix frequencies (*SfFreq*)

<sup>2</sup>Indian Institute of Information Technology and Management-Kerala

are also measured from word lists of all four languages. These measured values are shown in Table 3.

Language	<i>StCount</i>	<i>StFreq</i>	<i>SfCount</i>	<i>SfFreq</i>	<i>X</i>
English	4974	2.01	43	232.58	436
Hindi	4792	2.09	134	74.63	726
Marathi	4086	2.45	604	16.56	1958
Malayalam	1077	9.29	762	13.12	26248

Table 3: Statistics of Word List and Suffix List (*X*)

Number of distinct stems in the word form list decreases and average stem frequencies increases along with morphological complexity of language. Similarly the number of distinct suffixes in the word list increases and average suffix frequencies decreases along with morphological complexity. We can also see that the number of suffixes in a language also increases with morphological complexity. Since these patterns are quite intuitive, the data taken for experiments seems to be proper samples that represents the languages.

## 7.2 Results and Discussion

The accuracy of four baselines and two newly proposed systems for four languages are tabulated in Table 4. The results indicates the effectiveness of the new systems over baseline systems across various languages. Improvement in the performance of WMSS over MSS is also clearly visible in the table. Above 80% accuracies for all languages are obtained by using the WMSS model. English, Hindi, Marathi and Malayalam are the languages in the increasing order of morphological complexity. We can observe that the accuracies are decreasing along with the morphological complexity of language. This indicates stemming is difficult for morphologically complex languages.

Language	Random Stem	Largest Stem	Smallest Stem	Modified Linguistica	MSS	WMSS
English	44.98	47.39	49.36	53.82	84.44	88.86
Hindi	50.68	43.04	57.44	62.74	80.71	83.98
Marathi	41.66	30.44	69.766	59.33	78.28	80.19
Malayalam	19.31	3.51	57.58	65.86	78.32	80.06

Table 4: Stemming Accuracies in Percentage

For all four languages, the baseline which selects stems with maximum length have a lesser score than the baseline which selects stems with minimum length. This shows, if there are more

than one stem candidate, then smaller stems is preferred. Since null suffix is present in the suffix list, maximum stem length baseline always selects the word itself as its stem. So the low score for maximum stem length baseline for Malayalam indicates, most of the Malayalam words are in the inflected form. To get better insight about the remaining issues an error analysis of the output sample is required.

### 7.3 Error Analysis

To get better insight about the remaining issues, erroneous samples generated by the best performing system, *i.e.*, WMSS, are categorized in to under stemming<sup>3</sup>, over stemming<sup>4</sup> and weight error. The weight error is the case where the correct stem is in the word list but the identified incorrect stem is not. Such errors can be corrected by modifying the weight function in the WMSS formulation. Percentage of errors in various categories are tabulated in Table 5.

If the number of suffixes in the word list is very small compared to the total number of suffixes in the suffix list, then there is a high chance for overstemming. So the ratio between total number of suffixes and the number of suffixes in the word list (suffix ratio), for all four languages are also included in Table 5.

Language	Under-Stemming	Over-Stemming	Weight-Error	Suffix ratio
English	2.76	8.38	3.74	10.0
Hindi	3.90	12.12	5.94	5.42
Marathi	8.36	11.45	5.57	3.24
Malayalam	0.93	19.01	0.24	34.5

Table 5: Percentage of Errors and Suffix ratios

Suffix ratio of English is high (10). It decreases in Hindi, and further decreases in Marathi. According to this pattern, the under stemming errors are very few (only 3%) in English and it increases in Hindi and Marathi. The suffix ratio of Malayalam is higher than English so the under stemming errors are negligibly small (less than 1%). The relation between suffix ratio and under stemming errors are clearly visible from these numbers. So to reduce the under stemming errors, we need to increase the number of input suffixes.

<sup>3</sup>identified stem is longer than correct stem, *e.g.*, *mosse* in *mosses*

<sup>4</sup>identified stem is shorter than correct stem, *e.g.*, *s* in *sing*

The over stemming errors increases from English to Malayalam. This indicates that the over stemming errors are more sensitive to morphological complexity than the suffix ratio. From the table we can see that, weight errors are significant except in Malayalam. This indicates the requirement of weight modification. Also, we can see that the weight errors are high in Hindi and Marathi, and hence the weight modification is crucial for these languages.

After the analysis, the main observation is about the importance of weight modification. Some sample words from all four languages are shown in Figure 1.

Language	Word	Identified Stem	Correct Stem (in case of erroneous sample)
English	pretenses halfways rhodes.	pretenses halfway rhodes	pretense
Hindi	मच्छर (machhar) (mosquito) चूर्ण (choornom) सोफे (sophe)	मच्छ (machha) चूर्ण (choorn) सोफ (sofa)	मच्छर (machhar)
Marathi	छेडतील (chhedtheel) (to provoke) सांभाळतात (saambhaalthaath) (to look after) दिव्य (divya) (magnificent)	छेड (chhed) सांभाळ (sambhaal) दिव (diva)	दिव्य (divya)
Malayalam	കൂണിലുമുല്ലം (koonilumellam) (also in mushroom) പാമ്പിനുള്ളിൽ (paampinullil) (inside snake) അണുക്കൾ (anukkalum) (and atoms)	കൂണ (koona) പാമ്പ (paampa) അണ (ana)	അണു (anu)

Figure 1: Output Samples (WMSS)

## 8 Conclusion

Two algorithms for stemming, that produces a mapping from words to stems by minimizing the number of stems upto a limit, given a word list and a suffix list are proposed and implemented. Stemming systems that use these algorithms are evaluated using languages from Indo European and Dravidian families. Moderate to high accuracies of stemming are obtained in case of for all four languages: English, Hindi, Malayalam and Marathi.

Collecting a word list is relatively an easy task for a new language. But, collecting a complete list of suffixes is a much more involved task since detailed linguistic work is required. So completely unsupervised stemming is our future work. Stems will be produced from only the word form list.



## References

- V. Chvatal. 1979. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):pp. 233–235.
- Alexander Clark. 2001. Partially supervised learning of morphology with stochastic transducers. In *NL-PRS*, pages 341–348.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *TSLP*, 4(1).
- Sajib Dasgupta and Vincent Ng. 2006. Unsupervised morphological parsing of bengali. *Language Resources and Evaluation*, pages 311–330.
- Sajib Dasgupta and Vincent Ng. 2007. High-performance, language-independent morphological segmentation. In *HLT-NAACL*, pages 155–163.
- Markus Dreyer and Jason Eisner. 2009. Graphical models over multiple strings. In *Proc. of EMNLP-09*, pages 101–110.
- The British National Corpus, Version 3 BNC XML Edition. 2007. Distributed by Oxford University Computing Services on behalf of the BNC Consortium.
- Uriel Feige. 1998. A threshold of  $\ln n$  for approximating set cover. *JOURNAL OF THE ACM*, 45:314–318.
- Withrop N. Francis and Henry Kucera. 1964. *Manual of Information to accompany A standard corpus of present-day edited American English, for use with digital computers with Digital Computers*. Brown University Press.
- John A. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *CL*, (2):153–198.
- Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *CL*, (2):309–350.
- Harald Hammarström. 2006a. A naive theory of affixation and an algorithm for extraction. In *Proc. of HLT-NAACL-06*, pages 79–88, June.
- Harald Hammarström. 2006b. Poor man’s stemming: Unsupervised recognition of same-stem words. In *AIRS*, pages 323–337.
- David A. Hull. 1996. Stemming algorithms: A case study for detailed evaluation. *JASIS*, 47(1):70–84.
- Howard Johnson and Joel Martin. 2003. Unsupervised learning of morphology for english and inuktitut. In *Proc. of NAACL-HLT-03*, pages 43–45.
- Christian Monson, Jaime G. Carbonell, Alon Lavie, and Lori S. Levin. 2008. Paramor and morpho challenge 2008. In *CLEF*, pages 967–974.
- Matthew G. Snover, Gaja E. Jarosz, and Michael R. Brent. 2002. Unsupervised learning of morphology using a novel directed search algorithm: taking the first step. In *Proc. of ACL-WMPL-02*, pages 11–20.
- Antal van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *Proc. of ACL-99*.
- N. Vasudevan and Pushpak Bhattacharyya. 2012. Optimal stem identification in presence of suffix list. In *CICLing (1)*, pages 92–103.
- Eric W Weisstein. [Online; 30-04-2010]. Iverson bracket. MathWorldA Wolfram Web Resources. <http://mathworld.wolfram.com/IversonBracket.html>.

# What information is helpful for dependency based Semantic Role Labeling

Yanyan Luo    Kevin Duh    Yuji Matsumoto

Computational Linguistics, Nara Institute of Science and Technology

Takayama, Ikoma, Nara 630-0192, Japan

{yanyan-l; kevinduh; matsu}@is.naist.jp

## Abstract

Semantic Role Labeling (SRL) is an important task since it benefits a wide range of natural language processing applications. Given a sentence, the task of SRL is to identify arguments for a predicate (target verb or noun) and assign semantically meaningful labels to them. Dependency parsing based methods have achieved much success in SRL. However, due to errors in dependency parsing, there remains a large performance gap between SRL based on oracle parses and SRL based on automatic parses in practice. In light of this, this paper investigates what additional information is necessary to close this gap. Is it worthwhile to introduce additional dependency information in the form of N-best parse features, or is it better to incorporate orthogonal non-dependency information (base chunk constituents)? We compare the above features in a SRL system that achieves state-of-the-art results on the CoNLL 2009 Chinese task corpus. Our findings suggest that orthogonal information in the form of constituents is much more helpful in improving dependency based SRL in practice.

## 1 Introduction

In recent years, SRL has become an important component in many kinds of deep natural language processing applications, such as question answering (Narayanan and Harabagiu, 2004), event extraction (Riedel and McCallum, 2011), document categorization (Persson et al., 2009). SRL aims at identifying the semantic relations between predicates in a sentence and their associated arguments, with these relations drawn from a pre-specific list of possible semantic roles for

corresponding predicates. Syntax information is essential in SRL systems. To date, both constituent parsing and dependency parsing based SRL have been investigated (Xue, 2008; Johansson and Nugues, 2008), with dependency based systems giving superior results in CoNLL 2008 (Surdeanu et al., 2008) and CoNLL 2009 shared tasks (Hajič et al., 2009).

However, the performance gap is still quite large between SRL systems using oracle "perfect" dependency parses and SRL systems using automatic dependency parses. We observe as much as 10% F-score difference in our experiments. Clearly, errors in the 1-best dependency parse affects SRL prediction. This leaves an open question: in order to improve dependency based SRL, is it more worthwhile to incorporate more dependency information (in the form of N-best parse), or to incorporate an entirely separate source of information, such as base phrase chunks? We perform such an analysis in this paper, using a state-of-the-art Chinese SRL system.

Our findings suggest that constituent information such as chunking nicely complements dependency based SRL, achieving more improvements compared to N-best dependency information. Finally, we also report the best results to date on the CoNLL 2009 Chinese shared task.

## 2 Related Work

The bulk of previous work on automatic SRL has primarily focused on using full constituent parse of sentences to define argument boundaries and to extract relevant information for training classifiers. However, there have been some attempts at relaxing the necessity of using syntactic information derived from full parse trees. Sun et. al (2009) and Hacioglu et. al (2004) addressed the SRL problem on the basis of shallow syntactic information at the level of phrase chunks. In their approach, SRL is formulated as a sequence label-

ing problem, performing IOB2 decisions on the syntactic chunks of a sentence. However, this method ignores the full syntactic parsing information entirely, and we believe that even the accuracy of full syntactic parsing is not ideal, it is still helpful for SRL. Moreover, their method is inapplicable to dependency based SRL since a chunk usually consists of successive words.

A substantial amount of research has focused on dependency-based SRL (Meza-Ruiz and Riedel, 2009; Luo et al., 2012) since the CoNLL-2009 shared task and rich linguistic features (Zhao et al., 2009) are applied. For dependency related features, most studies focused on extracting them from the best dependency result. Johansson and Nugues (2008) tried to use N-best dependency parsing results. In their work, they applied 16-best dependency trees to generate predicate-argument structures and applied both syntactic trees and predicate-argument structures to a linear model. This model reranks the predicate-argument structures and the top 16 dependency trees at the same time. Though their work suggests that N-best dependency parsing can enhance the SRL, little is known about how the N-best dependency parsing related features perform on SRL.

### 3 Dependency based SRL Model

First, we define an instance as a predicate word and its corresponding argument words. If there are  $m$  predicates in a sentence, then there will be  $m$  instances. Given an instance  $X = \{x_1, \dots, x_p, \dots, x_n\}$  with the predicate position  $p$ , we want to find the corresponding sequence of argument labels and predicate sense  $S = a_1, a_{p-1}, P, a_{p+1}, \dots, a_n = \langle P, A \rangle$ . Each  $a_i$  for the  $i$ -th word in the instance  $X$  is drawn from a set of tags  $T(A)$  which contains all the semantic role labels in the corpus and which follows the definition criteria in Chinese PropBank. In addition, the special label *NONE* is added to  $T(A)$ . Words, labeled as *NONE*, are not arguments for the predicate. As for  $P$ , this is a member of a sense set  $T(x_p)$  which contains all possible senses of predicate word  $x_p$ . We propose two sorts of label assignment models  $Pr_{local}$  and  $Pr_{global}$ . The former can incorporate local features only; the latter can incorporate also global features. We use three types of local feature sets:  $F_P$ ,  $F_A$ ,  $F_{PA}$  and one global feature set  $F_G$ . These type definitions are the same as those in Watanabe et. al (2010).

### 3.1 Predicate Sense Disambiguation and SRL with a Local Model

Since the predicate cannot be an argument of itself for Chinese, we define the following local probabilistic model for argument classification and predicate sense disambiguation.

$$Pr_{local}(S|X) = \prod_{i=1(i \neq p)}^n Pr(a_i|P, X, i, p) \cdot Pr(P|X, p) \quad (1)$$

where  $Pr(a_i|P, X, i, p)$  and  $Pr(P|X, p)$  are estimated according to the following equation:

$$Pr(a_i|P, X, i, p) = \frac{1}{Z^A(X)} \exp\left\{ \sum_{f_{A_j} \in F_A} \lambda_{f_{A_j}} f_{A_j}(a_i, X) + \sum_{f_{PA_k} \in F_{PA}} \lambda_{f_{PA_k}} f_{PA_k}(a_i, X, p, P) \right\},$$

$$Pr(P|X, p) = \frac{1}{Z^P(X)} \exp\left\{ \sum_{f_{P_l} \in F_P} \lambda_{f_{P_l}} f_{P_l}(X, p, P) \right\},$$

where  $Z^A$  and  $Z^P$  are normalization functions, i.e.,

$$Z^A = \sum_{a_i \in T(A)} \exp\left\{ \sum_{f_{A_j} \in F_A} \lambda_{f_{A_j}} f_{A_j}(a_i, X) + \sum_{f_{PA_k} \in F_{PA}} \lambda_{f_{PA_k}} f_{PA_k}(a_i, X, p, P) \right\};$$

$$Z^P = \sum_{P \in T(x_p)} \exp\left\{ \sum_{f_{P_l} \in F_P} \lambda_{f_{P_l}} f_{P_l}(X, p, P) \right\};$$

$f$  are the features with associated weight  $\lambda$  learned via training.

### 3.2 Predicate Sense Disambiguation and SRL with the Global Model

Global information is known to be useful in SRL (Nakagawa, 2007). We propose a global probabilistic model  $Pr_{global}$  here for SRL as follows:

$$Pr_{global}(S|X) = \frac{1}{Z} Pr_{local}(S|X) \cdot \exp\left\{ \sum_{f_{G_m} \in F_G} \lambda_{f_{G_m}} f_{G_m}(S, X) \right\} \quad (2)$$

where  $Z$  is a normalizing factor over all candidate sequences  $S(X, p)$  (set of possible configurations of semantic tags and predicate senses given  $X$  and predicate location  $p$ ). To get the whole sequence of  $S$ , we need to perform a computationally expensive search. As done in previous work (Watanabe et al., 2010), we use a simple approach,

Type	%Error	#Error/#Occurrence
C	49.4%	7,162/14,497
G	88.62%	109/123
O	80.71%	3,175/3,934

Table 1: The distribution of SRL errors on development corpus by the joint model.

n-best relaxation. Unlike the  $Pr_{local}(S|X)$ , the product of probability distributions of each word, the probability distribution  $Pr_{global}(S|X)$  is calculated by feature functions  $f_G$  defined on an instance  $X$  with assignment  $S$ . Thereby, we can use any information in an instance without the independence assumption for assignments of words in it.

### 3.3 Error Analysis for Dependency-based SRL

Using the gold parse of dependency relations between a predicate and its arguments and according to these relations, we classified SRL errors into following three types.

- **C**: children of a predicate should be arguments but they are tagged incorrectly.
- **G**: grand children of a predicate should be arguments but they are tagged incorrectly.
- **O**: others

Table 1 shows the distribution of three errors observed in the development corpus after tagging by our joint model. For example, there are a total of 14,497 arguments that are children of predicates and among them, and 7,162(49.4%) are errors.

## 4 Results and Discussion

### 4.1 Experimental Setting

We used the Chinese dataset provided by CoNLL-2009 shared task for experiments. For comparison, two kinds of dependency parsing results are provided, the first is from MALT parser, the second is from second-order MST parser.

As for chunking information, we used the chunk definition presented in (Chen et al., 2006) to extract chunks from Chinese Tree Bank as training corpus. The line  $CH$  in Figure 1 shows the definition of chunks. In this example, "金融工作"(finance work) is a noun phrase and is composed by two nouns.

With the Inside/Outside representation for proper chunks and the following feature templates, where  $x_0$  is the current word, a CRF++<sup>1</sup> is trained for Chinese chunking task.

- Uni-gram word/POS tag features:  $x_{-2}$ ,  $x_{-1}$ ,  $x_0$ ,  $x_{+1}$  and  $x_{+2}$ .
- Bi-gram word/POS tag features:  $x_{-2}x_{-1}$ ,  $x_{-1}x_0$ ,  $x_0x_{+1}$  and  $x_{+1}x_{+2}$ .

### 4.2 Features

Most of features templates are "standard" which have been widely used in previous dependency-based SRL research (Johansson and Nugues, 2008; Luo et al., 2012). We do not explain "standard" features, however, we give a detailed description of the features used in this work.

#### 4.2.1 Base Phrase Chunking Related Features

In Figure 1, obviously, words in chunks do not have equal importance for SRL. Headwords represent the main meaning of the chunks. The base phrase chunking related features shown in Table 2 are only applied to these headwords. For other words in chunks, only lemma and POS information is used. The rules described in Sun and Jurafsky (2004) are used to extract headwords. Verb class in Table 2 is represented similarly as  $Verb.C1C2$ , which means this *verb* has two senses. For its first sense, it has one core argument and for its second sense, it has two core arguments. These verb classes are extracted from Chinese PropBank (Xue, 2008).

#### 4.2.2 Features from N-best Dependency Parsing

According to the statistics of development corpus, it is found that about 78.13% arguments are children of predicates. Even if its error percentage shown in Table 1 is less than 10%, the total error number is also considerable. If we can reduce the head errors for dependents, the  $C$  errors caused by dependency parsing errors should be decreased, and SRL tagging results would be improved. Under this hypothesis, we simply extracted the following features from every parse tree in the N-best list which are generated using second-order MST parser. These features are also included in the "standard" feature set when  $N = 1$ .

<sup>1</sup><http://crfpp.sourceforge.net/>

WORD	去年	西藏	金融	工作	取得	显著	成绩
POS	NN	NR	NN	NN	VV	JJ	NN
CH	[NP]	[NP]	[ NP ]	[VP]	[ADJP]	[NP]	
TAG	B-NP	B-NP	B-NP	I-NP	B-VP	B-ADJP	B-NP
SRL	TMP	NONE	NONE	A0	取得.01	NONE	A1

Figure 1: Chunking information for a predicate-argument structure.

Feature Name	Description
Chunk features	<p>chunk tag of headword with IB representation (e.g. <math>B - NP</math>)</p> <p>chunk tag of the chunk where the headword belongs to</p> <p>the number of words in a chunk</p> <p>the POS sequence of words in a chunk, for example, "金融工作" (finance work) is "NN_ NN"</p> <p>the position of the chunk with respect to the predicate(Position). There are three possible values: "before", "after" and "here".</p> <p>the conjunctions of Position and headword, predicate and verb class</p> <p>the conjunctions of Position and POS of headword, predicate and verb class</p> <p>lemma/POS of one word immediately before/after of the chunk</p>
Path features	<p>a chain of chunk types between the headword and the predicate.</p> <p>the length of the chunk chain between the headword and the predicate</p> <p>For example, chain of chunk types between headword "工作" and predicate "取得" is "NP-VP" and the length of the chunk chain is 2.</p>

Table 2: Chunking related feature template for experiments.

*Arguments'heads*: lemma/pos; lemma and pos; dependency label; whether they are predicates.

*Position*: position of the argument candidates with respect to the predicate positions in the tree; position of the heads of the argument candidates with respect to the predicate position in the sentence.

*Chain*: the left-to-right chain of the dependency labels of the predicate's dependents.

### 4.3 SRL Performance

The overall performance of SRL is calculated using the semantic evaluation metric of the CoNLL-2009 shared task scorer<sup>2</sup>. Table 3 gives the comparison of SRL performance before and after adding the proposed base phrase chunking related features on the test data. Lines with  $-/+$  show the SRL performance without/with base phrase chunking related features. As seen in this table, without gold dependency parse, the best SRL is up to 80.52 in  $F_1$  score. To the best of author's knowledge, there are few Chinese SRL results more than

<sup>2</sup><http://ufal.mff.cuni.cz/conll2009-st/eval09.pl>

	P(%)	R(%)	$F_1$ (%)
Gold parsing -	88.68	86.30	87.47
Gold parsing +	<b>90.03</b>	<b>87.71</b>	<b>88.86</b>
MALT -	82.64	72.68	77.34
MALT +	<b>84.17</b>	<b>74.67</b>	<b>79.13</b>
MST-2 -	83.01	75.39	79.02
MST-2 +	<b>84.49</b>	<b>76.92</b>	<b>80.52</b>

Table 3: SRL results without/with base phrase chunking information.

80%.

Although comparing the lines with  $-$ , it shows dependency parsing play the central role in Chinese SRL as expected. Comparing their corresponding lines with  $+$ , Chinese SRL can still benefit a lot from shallow parsing information. An example from the corpus is shown in Figure 2. Figure 2a shows the gold dependency parsing result and the gold predicate argument structure; Figure 2b shows the dependency parsing result from MALT parser and the predicate argument structure as a result of the predicted parse; Figure 2c shows the predicate argument structure which is predicted after adding base phrase chunking re-

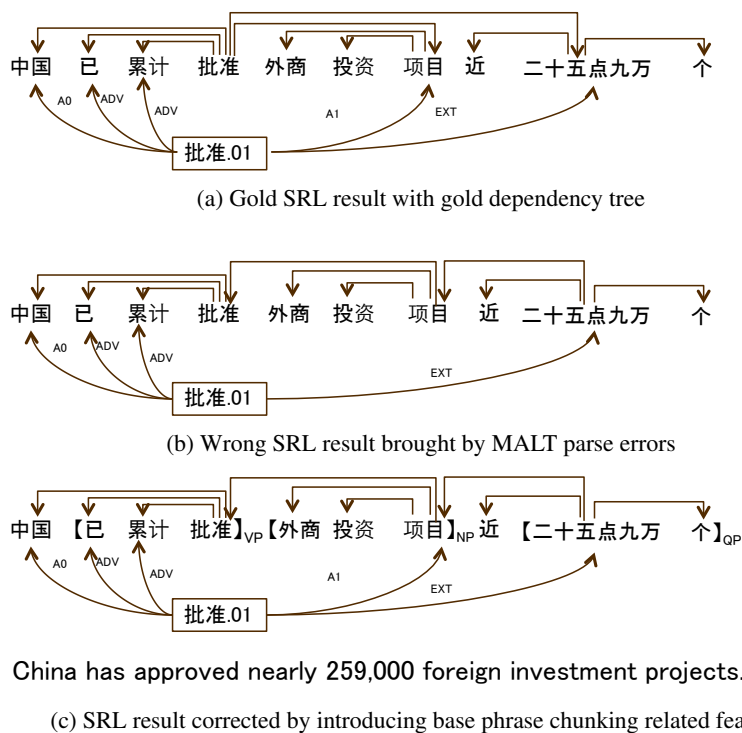


Figure 2: An example that the argument prediction error brought by MALT parse errors is corrected by introducing base phrase chunking related features.

	N-best	P (%)	R (%)	F <sub>1</sub> (%)
MST-2-	1	<b>83.01</b>	75.39	79.02
MST-2-	3	82.52	<b>77.16</b>	79.75
MST-2-	5	82.74	77.10	<b>79.82</b>
MST-2-	10	82.44	76.98	79.62

Table 4: SRL results with N-best dependency parsing related features.

N-best	Correct (#)	Error(#)	Noise(#)
1	18,428	3,176	-
3	19,071	2,533	4,636
5	19,392	2,212	5,667
10	19,738	1,866	7,699

Table 5: Dependency accuracy and the noise changes with different N.

lated features. In Figure 2c, the subscripts stand for chunk types. From Figure 2b, it can be seen that the argument A1 is not identified by the dependency based SRL because of dependency errors. Comparing Figure 2b and 2c, we can see that after adding the base phrase chunking related features, this SRL error brought by dependency parsing errors is corrected.

Line MALT+ and line MST-2- show that even the dependency parsing result from MALT is not better than that from second order MST, with the aid of chunking related features, Chinese SRL can still get comparable results.

Table 4 shows the Chinese SRL results after adding the N-best dependency parsing related features. It is not surprising that SRL can get better performance when  $N > 1$ , because the larger N, a more accurate dependency parsing results can be

likely obtained. When  $N = 5$ , SRL gets the best performance 79.82 in  $F_1$  with 0.8 point improvement.

However, the improvement declines when  $N = 10$ . A larger N may result in adding more accurate dependency parsing, however, it can also result in including more noises. For the MST parser using second order algorithm, Table 5 shows how the choice of the value of N affects the dependency parsing. The Correct(#) column represents the number of cases where the correct parent of an argument is predicted within the N-best. For example, in 3-best, it counts the number of arguments where their parents are correctly predicted in at least one of the 3 predictions. In the case where the parent is not predicted in any tree, they are counted as an error, as listed in the second column. The third column (Noise), is defined under

	N-best	P(%)	R (%)	F <sub>1</sub> (%)
[Björkelund, 2009]	-	82.42	75.12	78.60
[Meza-Ruiz, 2009]	-	82.66	73.36	77.73
[Zhao, 2009]	-	80.42	75.20	77.72
MST-2 +	1	<b>84.49</b>	76.92	80.52
MST-2 +	3	83.81	<b>78.51</b>	<b>81.07</b>
MST-2 +	5	83.71	78.40	80.97

Table 6: SRL results with base phrase chunking information and N-best parsing related features.

a hypothesis: correct dependency relations generate correct SRL results, wrong dependency relations generate incorrect SRL results. It represents the number of wrong dependency relations in Correct case which can cause bad influence on SRL results. For example, if 3 best heads for an argument are top-1, top-2, top-3 respectively, and top-1 is the correct one, then this case is a Correct case and the number of noise are 2; if none of the three results are correct, then this case is an Error case, and no noise. From this table, it obviously indicates that the benefit for dependency parsing brought by a larger N is less than the noise brought by the N.

With Tables 3 and 4, it can be seen that SRL benefits more from chunking related features than from N-best parse related features.

Table 6 shows the the results of Chinese SRL after adding base phrase chunking information and N-best parsing related features and gives the comparison with the previous work. From Tables 4 and 6 we can see that after adding the chunking related features, the impact of N-best parsing related features is a little reduced.

#### 4.4 Discussion

In Section 4.3, we see that both chunking and N-best parsing related features are helpful for Chinese SRL to some extent. In order to understand how they affect SRL, we analyze the results from three types of errors introduced in Section 3.3. Table 7 shows the error changes when different features are added.

Since accurate dependency information is not always available, the three types of errors should become larger when automatic dependency parsers are used. From Tables 1 and 7, the *C* and *O* errors increased as expected, while *G* de-

	N-best	C(%)	G(%)	O(%)
MST-2-	1	25.37	86.93	59.06
MST-2-	3	22.83	78.43	56.84
MST-2-	5	22.83	78.43	57.36
MST-2+	1	23.93	86.93	54.70
MST-2+	3	21.5	76.47	53.05
MST-2+	5	21.66	76.47	53.38

Table 7: SRL error changes with different features

creased. The main reason is that arguments, that are grandchildren of predicates, are relocated in the dependency trees because of dependency errors, and these locations make them easier to be tagged. From the first and fourth rows, they suggest that shallow parsing information are helpful to reduce the *C* and *O* errors. Comparing the fourth line with second and third rows, they explain why SRL achieves more improvements from chunking than from N-best dependency. When *N* changed from 1 to 3, the errors decreased obviously, however, when the *N* = 5, there are no obviously different changes.

## 5 Conclusions and Future Work

In this paper, we introduce additional information: base phrase chunking and N-best dependency parsing related features to a dependency based SRL system and investigate the benefit that our Chinese SRL model can get from them. Evaluations on the CoNLL 2009 Chinese corpus show that chunking information well complements dependency based SRL, achieving more improvements compared to N-best dependency information. With those additional features, our dependency based SRL achieves the best result on the same Chinese corpus to our knowledge. Furthermore, while all our experiments are for Chinese, it is possible to design experiments for other languages with our models.

Our experiment results show that we are not limited to increase SRL performance via more accurate syntactic parsing, but that we can explore other information, which is easier to get and is helpful for SRL. This also guides our future work. In our future work, we would like to explore more features and their influence on SRL.

## References

- Anders Björkelund, Love Hafdell and Pierre Nugues. 2009. Multilingual Semantic Role Labeling. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pp. 43-48.
- Wenliang Chen, Yujie Zhang and Hitoshi Isahara. 2006. An Empirical Study of Chinese Chunking. *Proceedings of the COLING/ACL on Main conference poster sessions*, pp. 97-104.
- Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin and Daniel Jurafsky. 2004. Semantic Role Labeling by Tagging Syntactic Chunks. *Proceedings of the 8th Conference on CoNLL-2004, Shared Task*.
- Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antòia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue and Yi Zhang. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pp. 1-18.
- Richard Johansson and Pierre Nugues. 2008. Dependency-based Semantic Role Labeling of PropBank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 69-78.
- Luo Yanyan, Asahara Masayuki and Matsumoto Yuji. 2012. Robust Integrated Models for Chinese Predicate-Argument Structure Analysis. *China Communications*, 9(3): pp. 10-18.
- Ryan McDonald, Koby Crammer and Fernando Pereira. 2005. Online Large-margin Training of Dependency Parsers. *Proceeding of the 43th Annual Meeting on Association for Computational Linguistics*, pp.91-98.
- Ivan Meza-Ruiz and Sebastian Riedel. 2009. Jointly Identifying Predicates, Arguments and Senses Using Markov Logic. *Proceedings of Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL-2009)*, pp. 155-163.
- Tetsuji Nakagawa. 2007. Multilingual Dependency Parsing Using Global Features. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, 34(2): pp. 952-956.
- Srini Narayanan and Sanda Harabagiu. 2004. Question Answering Based on Semantic Structures. *Proceeding of the 20th International Conference on Computer Linguistics*, pp. 693-701.
- Jacob Persson, Richard Johansson and Pierre Nugues. 2009. Fast and Robust Joint Models for Biomedical Event Extraction. *NODALIDA 2009 Conference Proceedings*, pp.142-149.
- Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text Chunking using Transformation-Based Learning. *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 88-94.
- Sebastian Riedel and Andrew McCallum. 2011. Fast and Robust Joint Models for Biomedical Event Extraction. *Proceeding of the 2011 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Languages Learning*, pp. 1-12.
- Honglin Sun and Daniel Jurafsky. 2004. Shallow Semantic Parsing of Chinese. *Proceedings of Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL-2004)*, pp. 1249-256.
- Weiwei Sun, Zhifang Sui, Meng Wang and Xin Wang. 2009. Chinese Semantic Role Labeling with Shallow Parsing. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 1475-1483.
- Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez and Joakim Nivre. 2008. The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic dependencies. *Proceedings of the 12th Conference on Computational Natural Language Learning*, pp. 157-177.
- Kristina Toutanova, Aria Haghighi and Christopher D. Manning. 2008. A Global Joint Model for Semantic Role Labeling. *Computational Linguistics*, 34(2): pp.161-191.
- Yotaro Watanabe, Masayuki Asahara and Yuji Matsumoto. 2010. A Structured Model for Joint Learning of Argument Roles and Predicate Senses. *Proceedings of the ACL 2010 Conference Short Papers*, pp. 98-102.
- Nianwen Xue. 2008. Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics*, 34(2): pp. 225-255.
- Hai Zhao, Wenliang Chen, Chunyu Kit and Guodong Zhou. 2009. Multilingual Dependency Learning: A Huge Feature Engineering Method to Semantic Dependency Parsing. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pp. 55-60.



# Classifying Taxonomic Relations between Pairs of Wikipedia Articles

**Or Biran**

Columbia University  
Department of Computer Science  
orb@cs.columbia.edu

**Kathleen McKeown**

Columbia University  
Department of Computer Science  
kathy@cs.columbia.edu

## Abstract

Natural language generation systems rely on taxonomic thesauri for tasks such as lexical choice and aggregation. WordNet is one such taxonomy, but it is limited in size. Motivated by the needs of a generation system in the scientific literature domain, we present a method for building a taxonomic thesaurus from Wikipedia articles, where each article represents a potential concept in the taxonomy. We propose framing the problem of creating a taxonomy as a classification task of the potential relations between individual Wikipedia article pairs, and show that a supervised algorithm can achieve high precision in this task with very little training data.

## 1 Introduction

Thesauri are useful resources for many NLP applications. In particular, taxonomic thesauri which contain synonymy and hypernymy relations are important for natural language generation (NLG) systems which must make decisions regarding lexical choice and aggregation. WordNet (Fellbaum, 1998) is one such thesaurus which has many uses in generation (Jing, 1998), but its set of concepts (called *synsets*) is quite limited. It does not contain many domain-specific concepts, nor does it contain technical concepts that emerged very recently. This work is motivated by the needs of a NLG system in the scientific literature domain, where these missing concepts are absolutely necessary for any practical application. Our goal is to generate a thesaurus containing synonymy and hypernymy relations between scientific terms which a generation system can use to select the most appropriate term given a context.

The English Wikipedia has over 4 million articles, and over 8.6 million titles if *redirects*, which

are alternative titles for the articles, are included. These titles are essentially lexical terms referring to concepts. Crucially, it contains articles describing many domain-specific concepts, and, in particular, scientific and technological concepts. For example, Wikipedia contains articles with titles such as *Supersymmetric String Theory*, *Gorilla Glass* and *Sentiment Analysis*, all of which are missing from WordNet. While there have been attempts to build ontologies from Wikipedia, these tended to focus (in their optimization and evaluation) on entities such as people, places and events. There is still a need for a WordNet-like taxonomy which would contain accurate synonymy and hypernymy relations for highly specialized terms from various scientific domains (for our purposes) and other specialized domains.

Unlike previous approaches, which tend to rely on WordNet's hierarchy and/or on Wikipedia's pseudo-hierarchy of *categories*, we frame the problem as a binary classification task for a pair of Wikipedia article titles - deciding whether the term representing the concept in the first article is a hypernym of the term representing the second or not. This enables us to handle specialized concepts which are far from the established concepts in the WordNet hierarchy.

WordNet-like taxonomies behave in some ways as a dictionary, in others as an ontology. To avoid confusion, we define the main terms we use in this paper and what they correspond to:

- A *concept* in computational ontologies is a unique semantic entity. We assume that WordNet synsets correspond to concepts. Another assumption we make is that each Wikipedia article describes something analogous to a concept; this assumption does not work for some types of articles (e.g. Template articles), and we remove such articles before processing, as explained in section 3.

- A *term* is a lexical entity (word or combination of words) used to refer to a concept. Each WordNet synset contains multiple terms (synonyms) which all refer to the concept represented by the synset. We treat Wikipedia article titles as terms referring to the concept described in the article. In addition to the main title, Wikipedia has multiple additional *redirect titles* referring to each article. We do not *a priori* treat these as synonyms, as they are often hypernyms, hyponyms or even terms referring to distinct (though related) concepts (for example, at the time of this publication, *Disambiguation* redirects to *Word Sense Disambiguation*; *nano-SIM* redirects to *Subscriber Identity Module (SIM)*; and *Sheep Sounds* redirects to *Sheep*).
- *Relations* in this work are semantic relations between pairs of terms - specifically, synonymy and hypernymy. This is in contrast to the use of the word in ontologies where relations occur between pairs of concepts.

The following are a few examples of relations that do not appear in WordNet and which our method correctly finds:

- *Gene Silencing* is a hypernym of *RNA Interference*
- *Graph Property* is a hypernym of *Clustering Coefficient*
- *Conditional Random Field* and *CRF* are synonyms

We will use these examples to illustrate the limitations of other methods in the next section.

## 2 Related Work

There have been many attempts to extend WordNet with concepts from Wikipedia. Because WordNet has some of the properties of an ontology, most work on extending WordNet with Wikipedia concepts was in the context of creating an ontology. Although our work is different in that we focus on extending only the taxonomic relations between the terms, this related work is still very relevant. There have also been attempts to create ontologies directly from Wikipedia in various ways, and we discuss those as well.

Yago (Suchanek et al., 2007) is a large ontology (over 10 million concepts) based on WordNet

and extended with concepts from Wikipedia and other resources. Its hypernymy hierarchy (a relation called *subClassOf*) is derived by matching articles with existing WordNet synsets using the lexical and syntactic properties of the title. This approach works well for some complex entities: a title like “American people in Japan” contains the head compound *people* which matches the WordNet synset *Person/Human*. It does not work as well for scientific concepts, where titles tend to be less clearly related. For example, Yago contains the concepts *Clustering Coefficient* and *RNA Interference*, because they are titles of Wikipedia articles; but these concepts are not part of the *subClassOf* hierarchy, because their titles are not lexically similar to *Graph Property* and *Gene Silencing*, respectively.

Ponzetto and Navigli (2009) link Wikipedia *categories* to existing WordNet synsets, leveraging the category structure to enrich WordNet with concepts from Wikipedia. Wikipedia categories are mostly thematic, with no strict hierarchical structure and do not represent a taxonomy, but they do tend to be somewhat hierarchical for concepts low in the hierarchy (i.e., more specific concepts). For example, *Public transport in Stockholm* is in the category *Public transport in Sweden* which is in the category *Public transport*, and the latter corresponds to a synset in WordNet. However, this is not true for many scientific concepts, where even the more general concept does not appear in WordNet. For example, *Clustering coefficient* is in the category *Graph invariants*, but the categories above that are purely thematic, and WordNet does not contain a synset for *Graph invariant*. Similarly, the term *CRF* is the title of a disambiguation page, which does not belong to any categories and so would not be linked to *Conditional Random Field*.

Syed and Finin (2010) match each Wikipedia article to a WordNet synset as a hypernym-like superclass. Their method relies on the synset-category mappings of (Ponzetto and Navigli, 2009), extending it with information obtained from the hyperlink structure of the Wikipedia articles. However, this approach is still limited by the choice of categories for each article. In addition, it does not work as well for articles with a small number of hyperlinks, which is typical of the more specialized scientific articles.

There have also been attempts (Auer et al.,

2007; Wu and Weld, 2008) to build ontologies from the *infoboxes* of Wikipedia articles, which commonly occur in articles of (e.g.) people and places but not in the articles of most domain-specific concepts.

There has also been work mapping words from Wikipedia articles to particular senses within WordNet using WSD techniques (Mihalcea, 2007; Milne and Witten, 2008). Our work is different in that we attempt to create a thesaurus specifically containing terms that are not in WordNet.

## 2.1 Contrast to Related Work

In addition to not being optimal for the scientific domain, these approaches all have in common that in attempting to extend WordNet using Wikipedia they rely on the structural information in WordNet directly. This generally means that the further down the hierarchy a term is (that is, the further it gets from the most specific hypernym available in WordNet) the less accurate the constructed taxonomy becomes with regard to its relations. This again works well for some entities, where WordNet contains reasonably specific concepts (e.g., occupations and nationalities for people, industries for organizations) but not too well for specialized concepts in specific domains.

In contrast, in our approach, WordNet is only used to provide the labels for very few relations (5,000) that are used in training and (separately) in evaluation. However, these relations are all considered individually. We do not rely on the WordNet hierarchical structure as a whole; instead, we learn to classify the relation between a pair of terms using only information from their Wikipedia article content. This makes our method more robust with regard to very specific concepts. Evaluating other methods using gold data from WordNet may be biased, because concepts from WordNet (even if they are not used directly in ontology construction) are inevitably close to other concepts in WordNet. It can be expected that for more highly specialized concepts, these methods will not perform as well. In our approach, there is nothing special about a relation whose concepts appear in WordNet, and performance on those should give a good indication of performance on other relations (perhaps with the caveat that concepts which appear in WordNet may have larger corresponding articles on average).

## 3 Data and Definitions

Since we want our terms from Wikipedia to refer to concepts, we remove from the Wikipedia corpus all the pages whose title begins with a wikipedia special prefix. These prefixes are single words followed by a colon, and denote a special type of wikipedia page, such as Template, Category or File. We also remove all pages whose title does not contain at least one English letter character.

We define a Wikipedia term as any Wikipedia article title and any redirect title which passes the filters above. This lexical definition is motivated by the need to find synonymy and hypernymy. It also makes evaluation (which we do using WordNet) more straightforward. To make things even simpler, we completely ignore senses. While word sense disambiguation has been a major part of some related work, it is less crucial for our purposes since specialized terms are less likely to be ambiguous than general terms. We hypothesize that the Wikipedia article itself describes the concept that is referred to by the term.

We define a WordNet term as any term (synonym) participating in any noun synset in WordNet. Wikipedia terms are matched to WordNet terms lexically, with some pre-processing: we lowercase the titles, replace underscores with spaces, remove diacritics from unicode characters and remove text in parentheses (which are commonly used in Wikipedia to disambiguate senses).

Using our definition, there are 117,092 WordNet terms. The total number of potential terms from Wikipedia is 9,096,022, which covers 73.62% of the WordNet terms. WordNet has 494,892 hypernym and synonym relations between all terms. The set of all potential relations from the Wikipedia term set (which is 9,096,022<sup>2</sup> in size) covers 63.71% of those.

We define our task as a binary classification over all potential relations from the Wikipedia term set. For each ordered pair of terms, we want to decide whether the first is a hypernym of the second or not. If two terms are determined to both be hypernyms of each other we treat them as synonyms. We evaluate on a dataset sampled from that subset of the Wikipedia terms which also exist in WordNet.

To determine the relations for all Wikipedia terms, the space of potential relations must first be dramatically reduced from its current size of over 82 trillion data points. In this paper, we present

results on sampled subsets.

## 4 Features

We extract fourteen features of four general types. For most of these, it is essential that each term in the pair corresponds to a Wikipedia article. Each term matches either the article title, or a redirect title that redirects to the article.

### 4.1 Features from the hyperlink structure of Wikipedia

We utilize the graph structure of hyperlinks between articles to build the following eight features:

1. First article links to second (yes or no)
2. Second article links to first (yes or no)
3. The cosine similarity between the outgoing links of the articles
4. The ratio of outgoing links in the first article shared by the second article
5. The ratio of outgoing links in the second article shared by the first article
6. The cosine similarity between the incoming links of the articles
7. The ratio of incoming links in the first article shared by the second article
8. The ratio of incoming links in the second article shared by the first article

One of the powerful aspects of Wikipedia is its hyperlink structure. Based on the simple assumption that article A links to article B only if the information in B is related to or somehow assists in understanding the information in A, the intuition is that two articles having a semantic relation will more often link to one another, and will in general link to more similar (additional) articles than will two unrelated articles. The Wikipedia hyperlink structure has been used to compute similarity between articles, for example in (Syed and Finin, 2010) and (Yazdani and Popescu-Belis, 2010).

Wikipedia links contain two bits of information: the title of the article they link to, and the text of the hyperlink as it appears in the referring article. For features (1) and (2), we allow both: that is, even if a hyperlink links to a third article, but uses the relevant article's title in the text,<sup>1</sup> we count that as a link to the relevant article. For the other features, we use only the title of the actual linked articles. The reason is that in features (1) and (2) we

<sup>1</sup>For example, a link for the article *New York City* may have only *New York* in the text, which is the title of an article about the state

want to measure something different than in the rest: whether or not one of the articles mentions the other directly (hyponyms often mention their hypernyms, while hypernyms sometimes list their hyponyms). An article being mentioned by name in a hyperlink, even when the link goes elsewhere, answers that criteria. The other features are intended to capture the similarity of the two articles based on how related the links to/from them are, and so using the text is less relevant (and that information would be captured to some extent by the feature in the next category instead).

### 4.2 Features from the text of the articles

For each article, we build a bag-of-words vector. These vectors are used to compute the cosine similarity between the two articles of a pair, which we use as a feature.

The intuition behind this central feature is that articles having a semantic similarity will also have a higher lexical similarity. This is the same intuition behind distributional similarity (Church and Hanks, 1990), which is that terms surrounded by similar context tend to be semantically related. In this case, the context does not surround the terms but is in the body of the articles corresponding to them. Lexical similarity between Wikipedia articles has been used successfully to link articles, for example in (Yazdani and Popescu-Belis, 2010).

### 4.3 Features from the redirect structure of Wikipedia

The Wikipedia dump contains a list of redirects from multiple alternative titles to each article. We use those to build three boolean features:

1. The first term redirects to the second term's article (yes or no)
2. The second term redirects to the first term's article (yes or no)
3. Both terms redirect to the same, third article (yes or no)

As mentioned earlier, redirect titles are often synonyms, hypernyms or hyponyms of the main title of the article they redirect to. While it is not consistent enough to use as a strict rule, this structure can be taken advantage of in features.

### 4.4 Features from the terms (i.e. the article titles)

In some cases, the terms themselves can point at the relation among them. In particular, hyper-

nyms are sometimes lexical subsets of their hypernyms (*String Theory* is a hypernym of *Super String Theory*; *Leukemia* is a hypernym of *lymphocytic leukemia* which in turn is a hypernym of *B-cell chronic lymphocytic leukemia*).

We therefore derive two features from the terms themselves (which correspond to article titles or redirect titles): the difference between the number of words in the two terms, and the number of words which overlap in the two terms.

## 5 Method and Evaluation

Our training, development and test data sets all consist of ordered pairs of terms from Wikipedia where both terms also appear in WordNet. The label is positive if the first term in the pair is a hypernym (or a synonym) of the second. The positive samples (which consist of pairs exhibiting either hypernymy or synonymy) are sampled from the relations in WordNet. To get negative samples we randomly pair terms from WordNet that have no relation between them.

We train two SVM classifiers: one on a small training set of 5,000 labeled pairs, and the other on a much larger set of 100,000 pairs. In both cases, the training sets are balanced and we used a balanced development set of 186,000 pairs. We then evaluate on a large unbalanced test dataset of 10 million pairs. Using the number of WordNet's total potential relations ( $117,092^2$ ) and the number of its true relations (494,892), we estimate the ratio of real relations in the natural set of all potential relations to be around 0.0036%. Estimating the factor by which we aim to reduce the size of the total space (of 82 trillion) as 1,000, the test set is then built using 360,000 sampled true relations from WordNet, while the rest are randomly paired concepts (which appear in WordNet but have no relation between them).

To illustrate our performance specifically on the science domain, we constructed a second data set using Wikipedia's category hierarchy. In this data set, we included only terms such that their corresponding articles are in a category which is a descendent of the *Science* category with a depth of no more than 20, but are *not* descendents of one of the following categories with a depth of 5 or less: *People*, *Places*, *History*, *Chronology*, *Music*, *Film* and *Sports*. These exclusions are required because descendents of the *Science* category include articles for entities such as scientists and

universities, certain historical dates/eras, and expansions of the technologies used in the music, film and sports industries to include entities from these fields (songs, bands, movies...) which then completely overwhelm the data set in size. The depth restrictions are necessary because the category graph is cyclic. In addition to illustrating performance in our intended domain, this test set is important in that it features negative samples that are not entirely random, since they are at least thematically related. The size of this set is 258,971, and it is unbalanced with about 10% positive samples. Note that we use the same classifier (trained on the same unrestricted training set) when evaluating on all test sets, including this one.

To illustrate our approach's advantage over naive methods, we include the results for two baselines. The first uses only the term names and makes predictions based on the Levenshtein distance between them (predicting synonym for distance  $< 8$ , hypernym for distance  $< 12$ , and none otherwise). The second predicts the relation type based on the lexical cosine similarity between the articles (predicting synonym for similarity  $> 0.1$ , hypernym for  $> 0.05$ , and none otherwise). The thresholds in both baselines were manually tuned to optimize f-measure on the development set.

In addition, we compare our performance with that provided by querying two leading publicly available ontologies that were constructed using Wikipedia's category hierarchy and infoboxes: Yago (Suchanek et al., 2007) and DBPedia (Auer et al., 2007).

We show two binary evaluations for each data set. The main evaluation, where a positive answer means the (ordered) pair has a hypernymy relation, is shown in Table 1. *SynonymOrNot*, in Table 2, is an additional evaluation over those pairs that were judged as having a relation in the first evaluation, and a positive answer means the pair is a synonym. Recall that we mark as synonyms those pairs that are determined to have both directional hypernyms. We found the results to be statistically significant using a standard t-test.

## 6 Discussion

The first thing to notice is that the SVM classifiers operate as high-precision, lower-recall systems for both tasks. On the *SynonymOrNot* task, precision is extremely high while retaining a reasonable recall even on the unbalanced test set. This is impor-

	Bal. P	Bal. R	Bal. F	Un. P	Un. R	Un. F	Sci. P	Sci. R	Sci. F
Naive baseline	57.41	69.44	62.85	4.76	69.38	8.91	13.74	80.08	23.45
Lexical baseline	97.14	17.89	30.21	54.31	16.23	24.99	70.22	19.13	30.06
DBPedia	100	0.25	0.5	96.33	0.26	0.52	98.72	1.78	3.5
Yago	100	15.23	26.44	99.96	14.5	25.33	100	29.19	45.19
SVM (trained on 5K samples)	98.75	46.18	62.93	66.03	42.95	<b>52.05</b>	64.81	61.23	<b>62.97</b>
SVM (trained on 100K samples)	98.13	48.46	<b>64.88</b>	57.51	45.22	50.63	57.45	66.3	61.56

Table 1: **Precision, Recall and F-measure** obtained for each data set for the main task. **Bal.** stands for **balanced**, the balanced development data set, **Un.** stands for **unbalanced**, the unbalanced test data set, and **Sci.** stands for **science**, the science-only filtered test set.

	Bal. P	Bal. R	Bal. F	Un. P	Un. R	Un. F	Sci. P	Sci. R	Sci. F
Naive baseline	50.76	68.61	58.35	7.02	66.19	12.7	7.65	64.26	13.67
Lexical baseline	68.41	97.83	<b>80.52</b>	43.49	97.75	60.2	23.99	92.31	38.08
SVM (trained on 5K samples)	99.92	30.15	46.33	99.65	44.58	<b>61.6</b>	97.65	56.12	71.28
SVM (trained on 100K samples)	99.92	29.92	46.05	99.62	44.46	61.48	97.75	58	<b>72.8</b>

Table 2: **Precision, Recall and F-measure** obtained for each data set for **SynonymOrNot**. **Bal.** stands for **balanced**, the balanced development data set, **Un.** stands for **unbalanced**, the unbalanced test data set, and **Sci.** stands for **science**, the science-only filtered test set.

tant, since a high precision is crucial to maintaining coherence in tasks such as lexical choice.

The classifiers beat both baselines on the main task. The lexical baseline does quite well on the *SynonymOrNot* task, but its performance deteriorates on the unbalanced test sets while the classifiers’ performance actually significantly increases due to its high-precision nature.

While the ontologies (Yago and DBPedia) offer incredibly high precision in all cases, their recall is very low (often less than 1% in DBPedia). This is because they focus on entities that are well defined through the category hierarchy and/or infoboxes, which most Wikipedia articles are not.

Overall, the classifiers beat both baselines and both ontologies in both tasks on both test sets. Most importantly, we achieve a relatively high performance on the science domain test set, which is our main goal in this paper.

Finally, it is interesting to note that there is little difference in performance between the SVM when trained on a small training set and when trained on a much larger training set. It seems that whatever can be learned about the data using these features (which is quite a bit, given the performance and especially the precision using this simple approach on a highly unbalanced test set) is learned very quickly, even from a small sampled set.

## 7 Conclusion and Future Work

We described a simple supervised method of classifying pairs of Wikipedia article titles in terms

of the relation among them, covering synonymy and hypernymy. Our approach significantly outperforms the baselines on simulated target data, and achieves very high precision. Unlike previously described approaches, it does not rely on the WordNet hierarchy as a whole, but only on the properties of the individual pair.

In order to use this method in building a taxonomic thesaurus from Wikipedia, we must first reduce the space of potential articles, which is tens of trillions in size. We leave this task and the task of building a full thesaurus to future work. Even without the full thesaurus, our approach can be used to make on-line decisions about the relation between any arbitrary pair of terms.

## Acknowledgments

This research is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

## References

- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.
- Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March.
- Quang Xuan Do and Dan Roth. 2010. Constraints based taxonomic relation classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1099–1109, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Hongyan Jing. 1998. Usage of wordnet in natural language generation. In *Proceedings of COLING-ACL'98 workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada, August.
- Rada Mihalcea. 2007. Using wikipedia for automatic word sense disambiguation. In Candace L. Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *HLT-NAACL*, pages 196–203. The Association for Computational Linguistics.
- David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA. ACM.
- Simone Paolo Ponzetto and Roberto Navigli. 2009. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *Proceedings of the 21st international joint conference on Artificial intelligence*, IJCAI'09, pages 2083–2088, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge - unifying WordNet and Wikipedia. In Carey L. Williamson, Mary Ellen Zurko, and Prashant J. Patel-Schneider, Peter F. Shenoy, editors, *16th International World Wide Web Conference (WWW 2007)*, pages 697–706, Banff, Canada. ACM.
- Zareen Syed and Tim Finin. 2010. Unsupervised techniques for discovering ontology elements from wikipedia article links. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 78–86, Los Angeles, California, June. Association for Computational Linguistics.
- Fei Wu and Daniel S. Weld. 2008. Automatically refining the wikipedia infobox ontology. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 635–644, New York, NY, USA. ACM.
- Majid Yazdani and Andrei Popescu-Belis. 2010. A random walk framework to compute textual semantic similarity: a unified model for three benchmark tasks. In *Proceedings of the 4th IEEE International Conference on Semantic Computing (ICSC 2010)*, Carnegie Mellon University, Pittsburgh, PA, USA, 0.

# A Rule System for Chinese Time Entity Recognition by Comprehensive Linguistic Study

**Hongzhi Xu**

The Department of CBS  
The Hong Kong Polytechnic University  
hongz.xu@gmail.com

**Chu-Ren Huang**

Faculty of Humanities  
The Hong Kong Polytechnic University  
churenhuang@gmail.com

## Abstract

Chinese time entity is quite complex. In this paper, we give a comprehensive linguistic study on it. Based on the analysis, we present a rule system which only considers the inner structure of Chinese time entities for the recognition. Experiments on Sinica and TempEval-2 corpus show that the rule system performs much better than the CRFs model. When using the rules as features within a CRFs model, the performance could be further improved.

## 1 Introduction

In SemEval-2010 competition, there is a sub task for temporal entity identification, which includes a Chinese corpus. The final goal of the task is to associate a temporal expression to a certain event. It is very important to extract all the elements for events in that it will be useful for event tracking. By identifying the time information of events will enable us to make inference on the temporal relation of different events.

In this paper, we will make a comprehensive study on Chinese time entities from a linguistic perspective and then present a rule system for recognizing them. Chinese temporal entity is very complex due to the flexible grammar of Chinese and the existence of many different time systems, such as Gregorian system, the Chinese lunar system, the Chinese tian-gan & di-zhi (GZ) time system.

Based on our linguistic analysis, we formalize a set of temporal elements that are the blocks used to construct time entities, such as *century*, *year*, *month*, *day*, *hour* etc. We then build a rule system that actually describe the topology of the temporal elements. For example, *year* follows *century*; *month* follows *year*. So, the model of our system is a directed graph, while a

valid temporal expression is a path from one certain node to another node. The longer the path is, the more confident the recognition will be.

CTEMP (Wu et al., 2005) also used linguistic rules for Chinese temporal entity recognition. However, the focus of this work differs from them in that we aims to identify Chinese time entities which could be described with a limited set of rules and can be easily translated into a structured format, such as TIMEX3(Pustejovsky et al., 2010) standard. For this part, the set of rules in this work are more comprehensive than (Wu et al., 2005). However, we don't include events that are used as time entities, since events intrinsically are not time entities. According to the Generative Lexicon Theory (Pustejovsky, 1995), this is a case of type coercion.

In Section 2, we will give a linguistic study on Chinese time entity expressions. In Section 3, we will construct a rule system which is mainly based on our linguistic study. In Section 4, we test rule system on Sinica and TempEval-2 corpora and give a discussion on the experimental result. Section 5 is the conclusion.

## 2 Chinese time entity: A linguistic study

We refer to Y.R. Chao's book (Chao, 1968) as a starting point of our study. In China, there are different time systems, including the lunar system, TianGan-DiZhi (GZ) system, etc. In ancient China, people used the emperor's reign to count time. When a new emperor appeared, a new period would then started.

In another perspective, people try to divide the time axis by different levels of granularity. Roughly, the whole axis can be divided into three periods: *guo-qu* (past), *xian-zai* (present) and *jiang-lai* (future). Smaller granularity includes century (*shi-ji*), year (*nian*), season (*ji-jie*), month (*yue*), day (*ri*), hour (*shi*), minute (*fen*), second (*miao*). Week (*zhou*) is a granularity that is independen-



t to year, season and month. In China, there are also jie-qi (JQ) that divides one year into 24 different periods. One month can also be divided into 3 periods (XUN): the first ten days (shang-xun), the second ten days (zhong-xun) and the left days (xia-xun). One day can also be divided into different vague phases (DP), e.g. before dawn (ling-chen), early morning (zao-shang), morning (shang-wu), noon (zhong-wu), afternoon (xia-wu), evening and night (wan-shang), midnight (wu-ye).

To compile rules for the automatic recognition of Chinese time entities, one important issue is to find out the construction regularity for each temporal element and the relations among the elements, which is also the inner structure of Chinese time entities.

## 2.1 Gregorian system and Chinese lunar system

Gregorian system starts from the year of Christ's birth. Before this year, B.C. (gong-yuan-qian) is used with a number to denote time on the time axis. After this year, A.D. (gong-yuan) is used, which is also the default value. Chinese supports this system. For example, 2013-08-08 09:01:01 is said in Chinese (gong yuan) *er-ling-yi-san-nian ba-yue ba-ri jiu-dian ling-yi-fen ling-yi-miao*.

One hour can also be divided into four quarters (ke). However, only *yi-ke* (fifteen) and *san-ke* (forty five) are valid expressions. For the half of an hour, *ban* (half) is used. *zheng* (right) will be used as the right start of an hour. So, *zheng, yi-ke, ban, san-ke* are the four possible values for the *KE* element.

One year can be divided into four quarters (ji-du:JD) or (ji-jie:season). An ordinal number will be used to refer to a certain JD, such as *di-yi ji-du* (the first quarter). The ordinal marker *di* could be omitted. So, *yi ji-du* is also a valid expression. Each season has its own name: spring (chun-ji), summer (xia-ji), autumn (qiu-ji) and winter (dong-ji).

For hours, day phases (DP) could be added before them. The DP is usually placed before *hour*, such as *ling-chen san-dian* (3:00am), *wu-ye shi-er-dian* (0:00). However, the boundaries of different phases are not clear, such as *xia-wu/wan-shang liu-dian* (6:00 in the afternoon/evening).

Century (shi-ji) can be followed by decade (nian-dai), such as *er-shi-shi-ji jiu-shi-nian-dai*

(the 90s of 20th century). The first decade is usually called *ling-ling-nian-dai* (00s) or *tou-shi-nian* (first ten years).

If *gong-yuan* (A.D.) or *gong-yuan-qian* (B.C) is used before *century* or *year*, then the numbers will be written as the pronunciation of the number rather than a sequence of digits. For example, *gong-yuan liang-qian-ling-yi-shi-san nian* is similar to be said as two thousand and thirteenth years A.D. in English. Otherwise, year 2013 will be written as *er-ling-yi-san-nian* (two-zero-one-three year).

Chinese lunar time system uses a similar way to denote time as the Gregorian system. However, it refers to the movement of the moon to count months. So the start of one year in lunar system is different from the Gregorian system. We can use a flag '&' (nong-li) to denote the lunar system, such as *& 2013-08-08*. In addition, the lunar system uses *chu* before the day number for the first ten days of a month in order to make up of two syllables, while the day marker *ri* is usually omitted. For example, Aug. 8th is said *ba-yue chu-ba*, Aug. 11th is said *ba-yue shi-yi*. The lunar label *nong-li* can also be placed before the subsequence of *year-month-day*, such as *nong-li wu-yue chu-wu (& 05-05)*, *nong-li chu-wu (& 05)*' etc.

## 2.2 TianGan-DiZhi system

This system was invented in Ancient China based on a the Chinese traditional philosophical theory. There are ten heavenly stems (tian gan: TG): *jia, yi, bing, ding, wu, ji, geng, xin, ren, gui* and twelve mundane branches (di zhi: DZ): *zi, chou, yin, mao, chen, si, wu, wei, shen, you, xu, hai*. Then, one year is denoted by a combination of two different elements circularly, which generates sixty different denotations. If we use a sequence number to denote the two elements, i.e.  $TG_{0-9}$  and  $DZ_{0-11}$ , then the  $i$ th year of a circulation is defined as  $y_i = TG_{i\%10}DZ_{i\%12}$ , where  $0 \leq i < 60$  and  $\%$  is the *mod* operation. For example, *gui-si-nian* (2013) can be formally denoted as year  $TG_9DZ_5$ , or simply  $GZ_{9,5}$ . Similarly, month, day and the Chinese hour can also be denoted like this.

The twelve DZ items are also associated with twelve animals (sheng xiao: SX): *shu* (mouse), *niu* (cattle), *hu* (tiger), *tu* (rabbit), *long* (dragon), *she* (snake), *ma* (horse), *yang* (sheep), *hou* (monkey), *ji* (chick), *gou* (dog), *zhu* (pig). So, one year can also be simplified as [*animal*] *nian*. For example,

year 2013 can be also called as *she-nian* (year of snake), or formally denoted as  $SX_5$ . However, this kind of expression can only be said alone. It can rarely be said with month and day, such as *\*she-nian wu-yue* (the 5th month of year of snake).

### 2.3 Jie-Qi

As we have mentioned, there are also twenty four Jie-Qi (JQ) within one year: *li-chun, yu-shui, jing-zhe, chun-fen, qing-ming, gu-yu, li-xia, xiao-man, mang-zhong, xia-zhi, xiao-shu, da-shu, li-qiu, chu-shu, bai-lu, qiu-fen, han-lu, shuang-jiang, li-dong, xiao-xue, da-xue, dong-zhi, xiao-han, da-han*. Every six JQs corresponds to and divide one season. The JQs are actually time words and included in Chinese dictionaries. *JQ* usually follows *year* element, such as *er-ling-yi-san-nian qiu-fen* (qiu-fen of 2013).

### 2.4 Regnal year system

Ancient Chinese people have seen a new emperor as a starting point of a new period. A number is used to count the following years after that year. The first year is called *yuan-nian*, the second year is called *er-nian* (2nd year), etc. For example, *QianLong yuan-nian* stands for the year when QianLong became the emperor. However, there are hundreds of emperors in the history of China, and many of them are not recorded at all. So, the list of emperors is hard to be complete. Usually, the most used regnal years refer to the Qing Dynasty.

### 2.5 Weekdays

Weekdays (*xing-qi*) are expressed by *xing-qi* plus a number from one to six. Sunday doesn't use seven, but *ri/tian* (day). Formally, they can be written as  $XQ_{0-6}$ . *xing-qi* is also called *zhou* (week) or *li-bai* (go to church) that is borrowed from religious activities. However, when we use *zhou*, Sunday cannot be said as *\*zhou-tian*. Week days are usually placed after *day* and before *hour* as a parenthesis, such as *2013-10-15 (Tuesday) 3:00pm*.

### 2.6 Festivals and Events

Some days or day sequences are named as festivals. Festivals are usually based on Gregorian system, such as the national day (*guo-qing*). In China, there are some festivals that are based on lunar system, such as the autumn day (*zhong-qiu*), which is & *08-16*. Some JQs are also regarded as festivals, especially when there are vacations for them, such

as *qing-ming festival*. Festivals are usually used independently to other temporal elements. Meanwhile, most of the festivals have been lexicalized and included in dictionaries.

Some festivals' dates are dynamic. For example, Thanksgiving is the fourth Thursday of November in the United States. For such festivals, we need to construct a function to automatically select a certain day in the year of context, e.g. *select(Thursday, 4, November, \$Year)*. From this point of view, we need to build ontology for translating festivals into the TIMEX3 standard.

An event can denote a time, such as *hun-qian* (before marriage), *shi fa dang tian* (the day when it happened) etc. Sometimes, a time operator can explicitly change the event into a time entity, such as *qian* (before), *hou* (after) etc. However, such expressions are hardly to be complete, and we don't deal with events in this system.

### 2.7 Referential time

The demonstrative, such as *zhe* (this) and *na* (that), can be placed before some temporal elements to form a referential time (ref). For example, *zhe-yi-nian* (this one year), *ben-shi-ji* (this century). The general pattern of such construction is  $[zhe/na]+[number]+[classifier]$ . There are also some lexicalized referential time expressions, such as *jin-nian* (this year), *ming-tian* (tomorrow) etc.

### 2.8 Durations

Duration is an interval of two time spots, i.e. the starting time and the ending time, connected by *dao/zhi* (to). *cong* (from) can also be placed in front. For example, *(cong) shi-yue shi-wu-ri dao shi-yue shi-qi-ri* (Oct. 15th - Oct. 17th). When there is only one temporal element in the starting and ending time, which means that their parent elements are the same, the first time marker can be omitted. For example, *shi-yue shi-wu-(ri) dao shi-qi-ri* (Oct. 15-17). Sometimes, only the length information is expressed, such as *liang-nian* (two years), which is made up of a Chinese number plus a classifier.

### 2.9 Period phases

When talking about a specific time period, we can refer to its different phases, e.g. its starting period (*chu-qi*), middle period (*zhong-qi*) and final period (*mo-qi/hou-qi*). Period is different from duration

in that duration emphasizes the length, while period not. So, '\*liang-nian chu-qi' (the start of two years) is an invalid expression.

### 3 A rule system for Chinese time entity recognition

So far, we have discussed 24 temporal elements: *century, decade, year, month, day, hour, minute, second, season, XUN, JQ, JD, DP, SX, lunar, GZyear, SXyear, GZmonth, GZday, GZhour, regnalyear, weekday, festival, periodphase*. Since festivals are lexical time expressions and it is hard to provide a complete list of festivals, we don't recognize festivals in this version. However, it is possible to build festival ontology which could be used to translate them into Gregorian calendar. We also add a limited set for the referential time expressions such as *jin-tian* (today), *ben-shi-ji* (this century) etc. This introduces 9 elements: *refcentury, refyear, refmonth, refday, refhour, refminute, refsecond, refJD, refdecade*.

The rule system is actually trying to describe the topological relations of the elements. The final model is a directed graph, containing 32 nodes and 50 edges. Table 1 shows a subset of the edges as demonstration. There are three different symbols in the rules. *A-B* means *B* follows *A*. *>* and *<* means 'stick to'. For example, *A > B* means *A* follows and depends on *B*. In other words, *A* cannot be used alone. *A <> B* means that they stick to each other. We should note that *<>* doesn't mean that they must appear together. For example, if there is another rule *A <> C*, then *A* can appear together either with *B* or *C*.

century - decade	refcentury - decade
year - jq	refyear - jq
year - jd	refyear - jd
year - month	refyear - month
month - xun	refmonth - xun
month - day	refmonth - day
hour<ke	gzhour<ke
lunar>year	lunar>month
lunar>gzyear	gzyear - gzmonth
day - dp	dp - hour
hour<minute	minute<second

Table 1: Topological Relation of Temporal Elements.

The recognition of time expressions includes two phases: identify the temporal elements and then concatenate the elements to get sequences based on the topological relations of them and the constraints described in Table 1. The recognition of temporal elements are implemented by regular expressions. The topological relation could be modeled as an acyclic graph.

#### 3.1 Convert to TIMEX3 format

In Chinese, the numbers in each temporal element can be a sequence of either Chinese or Arabic digits. For example, *er-ling-ling-san-nian* (year 2003) can also be written as *2003-nian*. For this kind of expressions, we need a parser to get the Chinese numbers first, which has been embedded in our system. Meanwhile, it can also parse them into machine readable integers.

In Chinese, we can also use Arabic numbers. In our system, we build a parser that could translate both Arabic and Chinese number into machine readable integers. However, due to the space limitation, we will not describe the parser here. Once we get numbers for each element. Some heuristic rules can be used to filter some false positive examples. For instance, *er-shi-san-dian* (23:00) is a legal time expression, while *er-shi-wu-dian* (25:00) is illegal. It appears in text because it can also mean (25 points). We add constraint on the value of *month*(1, 12), *day*(1, 31), *hour*(0, 24) etc.

Based on our rule system, the converting to TIMEX3 format is quite straightforward since the rules are based on the inner structures of Chinese time entities. In cases of referential temporal elements, such as *refyear, refday*, we can first place a variable for further processing, since the resolution of such references is an independent task. However, this will be our future work. For festivals, as we mentioned that most festivals have fixed date. So, a festival dictionary will be needed.

Nevertheless, translating time entities into machine readable format is a great advantage of rule systems. Even though static methods can give higher performance on recognition, there is no obvious way how to convert the time entities into machine readable format unless conversion rules are compiled, which then will resort to the inner structure of the entities which is then the work done by our rule system.

Corpus	#Words	#Entity
Sinica	10M	88K
TempEval-2 Training	23K	766
TempEval-2 Test	10K	191

Table 2: Corpus Information.

## 4 Experiments

We use two different corpora: Sinica (Chen et al., 1996) and TempEval-2 from SemEval-2010 competition (Pustejovsky and Verhagen., 2009). Sinica Corpus contains 10M words and the total number of time entity is 88K as shown in Table 2. The time words are tagged as ‘Nd’. However, there is no entity information. So, when an entity is recognized by our system, we first separate it into elements and then calculate the performance. Durations are labeled as *number + classifier* in Sinica, which are not time words. So, we don’t recognize durations in Sinica. For regnal year system, we only include a list of emperors of the Qing dynasty. We don’t deal with festivals as most of them are already lexicalized and are beyond the scope of entities. In other words, they can be recognized with a dictionary in a general word segmentation task.

TempEval-2 corpus includes training and test parts, as shown in Table 2 We analyse the annotation scheme based on training data and then add some additional rules on durations, such as *shinian* (ten years), *shi-tian* (ten days), and some approximate expressions, e.g. *shi-ji-nian* (more than ten years) and so on. Meanwhile, we add three new elements: *past* (guo-qu), *present* (xian-zai), *future* (jiang-lai). Each element includes a list of Chinese words.

### 4.1 Experimental results and Discussion

Table 3 shows the overall performance on Sinica and TempEval-2 corpora. Our rule system gives a high performance. Table 4 shows the precision and the number of recalled entities for some selected frequent patterns from 91 patterns identified from Sinica. Some long patterns give 1.0 precision. Some patterns are quite ambiguous, such as *hour-minute*. This is due to fact that *dian* means both the point in float numbers and time hour, and *fen* means both minute and score point in Chinese. For example, *san dian wu fen* means both 3:05 and 3.5 points. Regarding the different performances of different patterns, we can assign a confidence

Corpus	Precision	Recall	F1
Sinica	0.9429	0.8009	0.8661
TE-2-Train	0.9223	0.7898	0.8509
TE-2-Test	0.8876	0.8272	0.8564

Table 3: Performance of the rule system on time entity extraction.

Pattern	Prec.	#Rec.
month-xun	1.0	574
month-day-dp-hour	1.0	356
month-day-dp	1.0	315
regnalyear-month-day	0.9985	671
month-day	0.9963	7094
refday-dp	0.9957	2327
year-month-day	0.9931	2008
regnalyear-month	0.9918	363
refyear-month	0.9910	3098
day-dp-hour	0.9875	631
dp-hour	0.9855	1764
regnalyear	0.9836	1319
refyear-month-day	0.9831	1221
day-dp	0.9831	814
refday-dp-hour	0.9824	893
year-month	0.9819	1407
year-season	0.9775	261
refday-dp-hour-minute	0.9755	558
century-periodphase	0.9674	208
season	0.9658	2401
refday	0.9622	11670
refyear	0.9594	3706
century	0.9473	1384
month	0.9368	4119
dp-hour-minute	0.9336	633
year	0.9247	7324
decade	0.9148	569
weekday	0.9147	1458
day	0.8201	3592
hour-minute	0.8172	474
hour	0.6673	1073
refyear-periodphase	0.4740	219

Table 4: Matched patterns on Sinica corpus.

value to each pattern, such as the length of the extracted patterns plus F1-value on a training corpus. This will be helpful when incorporating the patterns into other systems.

Basically, the longer the matched pattern is, the more confident it is. However, as we can see that, some long patterns have a low precision. This is

Pattern	Prec.	#Rec.
year	1.0	133
month	1.0	8
year-month	1.0	4
decade-periodphase	1.0	6
refcentury-periodphase	1.0	5
refyear-firstnmonth	1.0	4
refday-dp	1.0	4
year-periodphase	1.0	5
refyear	0.9817	107
month-day	0.9783	45
refday	0.9412	32
refyear-periodphase	0.9	9
yearlength	0.875	56
day	0.8571	6
year-month-day	0.8571	6
present	0.848	106
past	0.625	15

Table 5: Performance of the rule system on TempEval-2 training corpus.

mainly due to the annotation errors that have split certain temporal elements into number-classifier construction in Sinica Corpus. For example, *er-shi-wu-ri* (the 25th) is annotated as *er-shi-wu* (25) plus *ri* (day).

Most ambiguous patterns contain one element, such as *year* and *day*. They can be both a date and a duration when the number is expressed in Chinese or Arabic digits. For example, *13 nian* (13 year: year 2013) could also be thirteen years. In Sinica, durations are labeled as *number + classifier*, which are not time words. In TempEval-2 corpus, both date and duration are entities. So, it will not be a problem for detection on this corpus. The ambiguity of such patterns introduced most of the false positive examples.

Table 5 and Table 6 show the identified patterns and their precision and the number of recalled entities. Compared to Sinica corpus, TempEval-2 corpus is quite sparse, and the element *refyear* such as *jin-nian* (this year) and *present* such as *mu-qian* (currently), take up a large part of the entities. This problem will affect the evaluation result in that the identification of time words e.g. *refyear* and *present* will be important to the overall performance.

In order to compare our rule system with the state-of-the-art statistical models. We also built a

Pattern	Prec.	#Rec.
refyear-month	1.0	9
month-xun	1.0	2
month-periodphase	1.0	3
year	1.0	14
refyear-jd	1.0	3
month-day	1.0	8
refyear-periodphase	1.0	18
refyear	1.0	20
present	0.9348	43
refyear-month	0.8571	6
future	0.8333	5
yearlength	0.625	10
refday	0.6	3
past	0.5714	4

Table 6: Performance of the rule system on TempEval-2 test corpus.

Type	Feature
Context	$token_{-1}$ , $token_0$ , $token_1$ , $token_{-1}+token_0$ , $token_0+token_1$
NGram	$unigram\_of\_token$ , $bigram\_of\_token$ , $trigram\_of\_token$
Structure	$end\_with\_classifier$ , $start\_with\_number$ , $number + classifier$

Table 7: Features used in CRFs model.

CRFs classifier with CRF++<sup>1</sup> on TempEval-2 corpus. The features used are shown in Table 7. To study whether the rule system could help the statistical model, we also use the recognition results of our rule system as pattern features. The result is shown in Table 8. We can see that the rule system gives a much higher performance than CRFs without using the patterns as features, i.e. 0.8564 v.s. 0.7787.

we also conduct experiment to test the statistical model based on characters with features shown in Table 9. This setting is actually more reasonable than word based, since word segmentation and entity recognition are overlap tasks. The result is shown in Table 10. We can see that, compared to word based setting, the performance increased

<sup>1</sup><http://crfpp.googlecode.com/svn/trunk/doc/index.html>

Feature	Precision	Recall	F1
Context	0.7699	0.4555	0.5724
+Structure	0.7867	0.6178	0.6921
+NGram	0.8373	0.7277	0.7787
+Pattern	0.8941	0.7958	0.8421
Rule System	0.8876	0.8272	0.8564

Table 8: Performance of time entity extraction with CRFs on TempEval-2 corpus.

Type	Feature
Context	$char_{-1}$ , $char_0$ , $char_1$ , $char_{-1}+char_0$ , $char_0+char_1$
Structure	$is\_number$ , $is\_classifier$ ,

Table 9: Features used in CRFs model based on characters.

CRFs	0.8476	0.7277	0.7831
+Pattern	0.8977	0.8272	0.8610
Rule System	0.8876	0.8272	0.8564

Table 10: Performance of time entity extraction with CRFs based on characters on TempEval-2 corpus.

from 0.7787 to 0.7831. This may due to the fact that with the segmentation information, the context features will be more sparse. When combining the patterns in CRFs model, the performance could be slightly improved. Overall, we can say that the inner structure of Chinese time entity is more important than context features.

**The false negative examples of the rule system in Sinica** includes some patterns that are not included in our system, some of which we think is not normal constructions of time expressions. For example, an Arabic digit sequence without the year marker *nian*, such as 2013, is also possibly a year element. Another one is the regnal year pattern, i.e. the *min-guo* period established in 1912 after Qing dynasty. However, there are many examples like *ba-shi-ba-nian* (88th years) with *min-guo* omitted.

**The false negative examples of the rule system in TempEval-2** includes some time word-

s that are not encoded in the rules. Some entities contains weekdays as a parenthesis, such as *jin-ri (xing-qi-er)* meaning *today (Tuesday)* will be treated as two entities. Some durations such as *san-[pause punctuation]-wu-nian* (three to five years). These are also not included in our system. The bare-number year is also a problem in this corpus.

## 5 Conclusion

In this paper, we made a linguistic study on Chinese time entities and presented a rule system for automatic recognition. We compare our system with CRFs model and the experiments on two different corpora showed that it gave a higher performance than the baseline system based on a CRFs model. When combining the rules with CRFs, the performance could be improved.

## Acknowledgments

The work is supported by a General Research Fund (GRF) sponsored by the Research Grants Council (Project no. 543810 and 544011).

## References

- Yuen Ren Chao. 1968. *A Grammar of Spoken Chinese*. Berkeley: University of California Press.
- Keh-Jiann Chen, Chu-Ren Huang, Li-Ping Chang, and Hui-Li Hsu. 1996. Sinica corpus: Design methodology for balanced corpora. In *Proceedings of Pacific and Asian Conference on Language and Information Computation*, pages 167–176.
- James Pustejovsky and Marc Verhagen. 2009. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions.*, pages 112–116.
- James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 18–21.
- James Pustejovsky. 1995. *The Generative Lexicon*. Cambridge: The MIT Press.
- Mingli Wu, Wenjie Li, Qin Lu, and Baoli Li. 2005. Ctemp: A chinese temporal parser for extracting and normalizing temporal information. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP05)*, pages 694–706.

# Financial Sentiment Analysis for Risk Prediction

Chuan-Ju Wang<sup>†</sup>, Ming-Feng Tsai<sup>†</sup>, Tse Liu<sup>†</sup>, Chin-Ting Chang<sup>‡</sup>

<sup>†</sup>Department of Computer Science &  
Program in Digital Content and Technology  
National Chengchi University  
Taipei 116, Taiwan  
{mftsai, g10120}@cs.nccu.edu.tw

<sup>‡</sup>Department of Computer Science  
University of Taipei  
Taipei 100, Taiwan  
cjiang@utaimei.edu.tw  
g10116006@go.utaimei.edu.tw

## Abstract

This paper attempts to identify the importance of sentiment words in financial reports on financial risk. By using a finance-specific sentiment lexicon, we apply regression and ranking techniques to analyze the relations between sentiment words and financial risk. The experimental results show that, based on the bag-of-words model, models trained on sentiment words only result in comparable performance to those on origin texts, which confirms the importance of financial sentiment words on risk prediction. Furthermore, the learned models suggest strong correlations between financial sentiment words and risk of companies. As a result, these findings are of great value for providing us more insight and understanding into the impact of financial sentiment words in financial reports.

## 1 Introduction

Sentiment analysis is the task of finding the attitudes of authors about specific objects. In recent years, because of the explosion of sentiment information from social web sites (i.e., Twitter and Facebook), blogs, and online forums, sentiment analysis has become one of the popular research areas in computational linguistics, such as (Narayanan et al., 2009; Mohammad and Turney, 2010).

The growing importance of Sentiment Analysis applied to finance brings forth many research and practical issues to minds like “Why Sentiment Analysis is important?” In finance, there have been several studies (Loughran and McDonald, 2011; Price et al., 2012; Garca, 2013) using textual analysis to examine the sentiment of numerous news items, articles, financial reports, and tweets about public companies. Then, the examined sentiments can be used to reflect the correlations with other fi-

ancial measures, such as stock returns and volatilities. For most sentiment analysis algorithms, as mentioned in (Feldman, 2013), the sentiment lexicon is the most important resource. In (Loughran and McDonald, 2011), the Harvard Psychosociological Dictionary, a common dictionary for general sentiment analysis, is extended to be a finance-specific sentiment lexicon.

In this study, we attempt to use the finance-specific sentiment lexicon to model the relations between sentiment information and financial risk. In specific, we formulate the problem as two different prediction tasks: regression and ranking. For the regression task, we aim to use sentiment information to predict a company’s future risk, which is usually characterized by its real-value volatility. Instead of predicting the real-value volatility, in the ranking task, we try to employ sentiments to rank companies according to their relative risk levels. From the two tasks, we observe that, trained on the finance-specific sentiment lexicon only, both the regression models and ranking models can obtain comparable performance to those trained on original texts, even though the word dimension is largely reduced from hundreds of thousands to only one and half thousand. In addition, we also conduct some analyses on the learned models, which can provide more insight into the financial sentiments.

The remainder of this paper is organized as follows. Section 2 introduces the financial risk measure and describes the problem formulations. In Section 3, we describe the details of our experimental settings and then report the experimental results. Some discussions and analyses on the learned models are provided in Section 4. Section 5 concludes.

## 2 Methodology

### 2.1 Stock Return Volatility

In finance, volatility is a common risk metric measured by the standard deviation of a stock’s returns

over a period of time. Let  $S_t$  be the price of a stock at time  $t$ . Holding the stock for one period from time  $t - 1$  to time  $t$  would result in a simple net return:  $R_t = S_t/S_{t-1} - 1$  (Tsay, 2005). Therefore, the volatility of returns for a stock from time  $t - n$  to  $t$  can be defined as follows:

$$v_{[t-n,t]} = \sqrt{\frac{\sum_{i=t-n}^t (R_i - \bar{R})^2}{n}}, \quad (1)$$

where  $\bar{R} = \sum_{i=t-n}^t R_i / (n + 1)$ .

## 2.2 Financial Sentiment Lexicon

For most sentiment analysis algorithms, a sentiment lexicon is the most crucial resource. As mentioned in (Loughran and McDonald, 2011), a general purpose sentiment lexicon might misclassify common words in financial texts. As shown in their paper, almost three-fourths of the words in the 10-K financial reports from year 1994 to 2008, which are identified as negative by the widely used Harvard Psychosociological Dictionary, are typically not considered negative in financial contexts.

In this paper, we use a finance-specific lexicon that consists of the 6 word lists provided by (Loughran and McDonald, 2011) to analyze the relations between these sentiment words and financial risk. The six lists are shown as follows:<sup>1</sup>

1. Fin-Neg: negative business terminologies (e.g., *deficit*, *default*).
2. Fin-Pos: positive business terminologies (e.g., *achieve*, *profit*).
3. Fin-Unc: words denoting uncertainty, with emphasis on the general notion of imprecision rather than exclusively focusing on risk (e.g., *appear*, *doubt*).
4. Fin-Lit: words reflecting a propensity for legal contest or, per our label, litigiousness (e.g., *amend*, *forbear*).
5. MW-Strong (Strong Modal Words): words expressing strong levels of confidence (e.g., *always*, *must*).
6. MW-Weak (Weak Modal Words): words expressing weak levels of confidence (e.g., *could*, *might*).

<sup>1</sup>All these lists are available at [http://www.nd.edu/mcdonald/Word\\_Lists.html](http://www.nd.edu/mcdonald/Word_Lists.html).

## 2.3 Problem Formulation

### 2.3.1 Regression Task

Given a collection of financial reports  $D = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n\}$ , in which each  $\mathbf{d}_i \in \mathbb{R}^p$  and is associated with a company  $c_i$ , we seek to predict the company's future risk, which is characterized by its volatility  $v_i$ . Such a prediction can be defined by a parameterized function  $f$  as follows:

$$\hat{v}_i = f(\mathbf{d}_i; \mathbf{w}). \quad (2)$$

The goal is to learn a  $p$ -dimensional vector  $\mathbf{w}$  from the training data  $T = \{(\mathbf{d}_i, v_i) | \mathbf{d}_i \in \mathbb{R}^p, v_i \in \mathbb{R}\}$ .

Support Vector Regression (SVR) (Drucker et al., 1997) is a popular technique for training such a regression model. SVR is trained by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} V(\mathbf{w}) &= \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle \\ &+ \frac{C}{n} \sum_{i=1}^n \max(|v_i - f(\mathbf{d}_i; \mathbf{w})| - \epsilon, 0), \end{aligned}$$

where  $C$  is a regularization constant and  $\epsilon$  controls the training error. More details about SVR can be found in (Schölkopf and Smola, 2001).

### 2.3.2 Ranking Task

For the ranking task, our goal is to rank companies by using their financial reports according to the volatilities of stock returns. Following the work in (Tsai and Wang, 2013), we split the volatilities of company stock returns within a year into different risk levels, which can be considered as the relative difference of risk among the companies.

After classifying the volatilities of stock returns (of companies) into different risk levels, the ranking task can be defined as follows: Given a collection of financial reports  $D$ , we aim to rank the companies via a ranking model  $f : \mathbb{R}^p \rightarrow \mathbb{R}$  such that the rank order of the set of companies is specified by the real value that the model  $f$  takes. In specific,  $f(\mathbf{d}_i) > f(\mathbf{d}_j)$  is taken to mean that the model asserts that  $c_i \succ c_j$ , where  $c_i \succ c_j$  means that  $c_i$  is ranked higher than  $c_j$ ; that is, the company  $c_i$  is more risky than  $c_j$ . In this paper, we adopt Ranking SVM (Joachims, 2006) for the ranking task.

## 3 Experiments

This section first describes the details of our experimental settings. Then, we report the experimental results of the models trained on the finance-specific



Year	# of Documents	# of Unique Terms
1996	1,406	19,613
1997	2,260	26,039
1998	2,461	29,020
1999	2,524	30,359
2000	2,424	30,312
2001	2,596	32,292
2002	2,845	38,692
2003	3,611	48,513
2004	3,558	50,674
2005	3,474	53,388
2006	3,306	51,147

Table 1: Statistics of the Corpora.

Dictionary	# of Words	# of Stemmed Words
Fin-Neg	2,349	918
Fin-Pos	354	151
Fin-Unc	291	127
Fin-Lit	871	443
MW-Strong	19	10
MW-Weak	27	15
Total	3,911	1,664

Table 2: Statistics of the Financial Lexicon.

sentiments only and those on original texts for the regression and ranking tasks.

### 3.1 Experimental Settings

#### 3.1.1 Corpora and Preprocessings

In the United States, the federal securities laws require publicly traded companies to disclose information on a regular basis. A Form 10-K, an annual report required by the Securities and Exchange Commission (SEC), provides a comprehensive overview of the company’s business and financial conditions, and includes audited financial statements. In this paper, the 10-K Corpus (Kogan et al., 2009) is used to conduct our experiments, in which only Section 7 “management’s discussion and analysis of financial conditions and results of operations” (MD&A) is used because the section contains the most important forward-looking statements about the companies.

For the preprocessing, in our experiments, all documents and the 6 financial sentiment word lists were stemmed by the Porter stemmer, and some stop words were also removed. Table 1 lists the statistics of documents and unique terms in each year. Table 2 shows the statistics before and after

stemming in each of the 6 financial word lists. Note that some words occur in more than one word list, so the number of unique stemmed sentiment words is 1,546 rather than 1,664.

In addition, the twelve months before/after the report volatility for each company (denote as  $v^{-(12)}$  and  $v^{+(12)}$ , respectively) can be calculated by Equation (1), where the price return series can be obtained from the Center for Research in Security Prices (CRSP) US Stocks Database. For the ranking task, in order to obtain the relative risks among companies, we categorize the companies of each year into 5 risk levels by following the work in (Tsai and Wang, 2013).

#### 3.1.2 Feature Representation

In our experiments, for the bag-of-words model, two word features are used to represent the 10-K reports. Given a document  $\mathbf{d}$ , two word features (i.e., TFIDF and LOG1P) are calculated as follows:

- $\text{TFIDF}(t, \mathbf{d}) = \text{TF}(t, \mathbf{d}) \times \text{IDF}(t, \mathbf{d}) = \text{TC}(t, \mathbf{d})/|\mathbf{d}| \times \log(|D|/|\mathbf{d} \in D : t \in \mathbf{d}|)$ ,
- $\text{LOG1P} = \log(1 + \text{TC}(t, \mathbf{d}))$ .

Above,  $\text{TC}(t, \mathbf{d})$  denotes the term count of  $t$  in  $\mathbf{d}$ ,  $|\mathbf{d}|$  is the length of document  $\mathbf{d}$ , and  $D$  denotes the set of all documents in each year. Note that IDF is computed from the documents in a single year because the document frequency of a specific word may vary across different years. Following (Kogan et al., 2009), we also use the logarithm of the twelve months before the report volatility (i.e.,  $\log v^{-(12)}$ ) as an additional feature. We denote these trained models as TFIDF+ and LOG1P+ hereafter.

#### 3.1.3 Evaluation Metrics

For the regression task, the performance is measured by the Mean Squared Error (MSE) between the predicted ( $\hat{v}_i^{+(12)}$ ) and true log-volatilities ( $v_i^{+(12)}$ ).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \left( \log \left( v_i^{+(12)} \right) - \log \left( \hat{v}_i^{+(12)} \right) \right)^2,$$

where  $n$  is the number of tested companies.

For the ranking task, two rank correlation metrics are used to evaluate the performance in our experiments: Spearman’s Rho (Myers and Well, 2003) and Kendall’s Tau (Kendall, 1938). Given two ranked lists  $X = \{x_1, x_2, \dots, x_n\}$  and  $Y =$

Task (Features)		2001	2002	2003	2004	2005	2006	Mirco-avg
Regression (LOG1P+)	Mean Squared Error							
	ORG	<b>0.18082</b>	0.17175	0.17157	0.12879	0.13038	0.14287	0.15271
	SEN	0.18506	<b>0.16367</b>	<b>0.15795</b>	<b>0.12822</b>	<b>0.13029</b>	<b>0.13998</b>	<b>0.14894</b>
Ranking (TFIDF+)	Kendall's Tau							
	ORG	0.62173	<b>0.63626</b>	0.58528	<b>0.59350</b>	0.59651	0.57641	0.59965
	SEN	<b>0.63349</b>	0.62280	<b>0.60527</b>	0.59017	<b>0.60273</b>	<b>0.58287</b>	<b>0.60458</b>
	Spearman's Rho							
	ORG	0.65271	<b>0.66692</b>	0.61662	<b>0.62317</b>	0.62531	0.60371	0.62939
SEN	<b>0.66397</b>	0.65303	<b>0.63646</b>	0.61953	<b>0.63133</b>	<b>0.60999</b>	<b>0.63403</b>	

Table 3: Experimental Results of Using Original Texts and Only Sentiment Words.

$\{y_1, y_2, \dots, y_n\}$ ,

$$\text{Rho} = 1 - \frac{6 \sum (x_i - y_i)^2}{n(n^2 - 1)},$$

$$\text{Tau} = \frac{\#\text{concordant pairs} - \#\text{discordant pairs}}{0.5 \cdot n \cdot (n - 1)}.$$

For the measure of Kendall's Tau, any pair of observations  $(x_i, y_i)$  and  $(x_j, y_j)$  is concordant if the ranks for both elements agree; that is, if both  $x_i \succ x_j$  and  $y_i \succ y_j$  or if both  $x_j \succ x_i$  and  $y_j \succ y_i$ . In contrast, it is discordant if  $x_i \succ x_j$  and  $y_j \succ y_i$  or if  $x_j \succ x_i$  and  $y_i \succ y_j$ . If  $x_i = x_j$  or  $y_i = y_j$ , the pair is neither concordant nor discordant.

### 3.1.4 Parameter Settings

For the regression task, linear kernel is adopted with  $\epsilon = 0.1$  and the trade-off  $C$  is set to the default value of SVM<sup>light</sup>,<sup>2</sup> which are the similar settings to those in (Kogan et al., 2009). For ranking, linear kernel is adopted with  $C = 1$ , all the other parameters are set as the default values of SVM<sup>Rank</sup>.<sup>3</sup>

## 3.2 Experimental Results

Table 3 tabulates the experimental results, in which the training data is composed of the financial reports in a five-year period, the following year of which is the test data. For example, the reports from year 1996 to 2000 constitute a training data, and the learned model is tested on the reports of year 2001.

We compare the performance of the models trained on the original texts (denoted as ORG hereafter) with those on only sentiment words (denoted

as SEN hereafter). In our experiments, the word feature LOG1P is chosen for the regression task and TFIDF for the ranking one, as suggested in (Kogan et al., 2009) and (Tsai and Wang, 2013). Note that in these two studies, their models are trained on the original texts and the results are listed in the row denoted as ORG in Table 3. The bold face number in the table denotes the best result between ORG and SEN. As shown in the table, for the two tasks, the results of using only sentiment words, in most cases, perform better than those of using the original texts.

## 4 Analysis

### 4.1 Ranking vs. Regression

Figure 1 shows the top 10 learned words from both the ranking (TFIDF+) and regression (LOGP+) models trained on sentiment words only (SEN); in addition, the figure also lists the accumulated numbers of these words appearing in the 6 corresponding regression or ranking models.

Observe that the words learned from the ranking models are much more consistent than those from the regression ones. For example, the words "amend," "deficit," "forbear" appear in all of the 6 ranking models; in addition, there are 7 words from the ranking models get the majority vote with more than 4 occurrences, whereas only 3 words from the regression ones occur more than 4 times. On the other hand, there are 11 words from the ranking models and 20 words from the regression ones that occur only one time. The results shown in Figure 1 correlate with the findings in (Tsai and Wang, 2013), which states that adopting the ranking models to analyze the relations between financial risk

<sup>2</sup><http://svmlight.joachims.org/>

<sup>3</sup>[http://www.cs.cornell.edu/people/tj/svm\\_light/svm\\_rank.html](http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html)

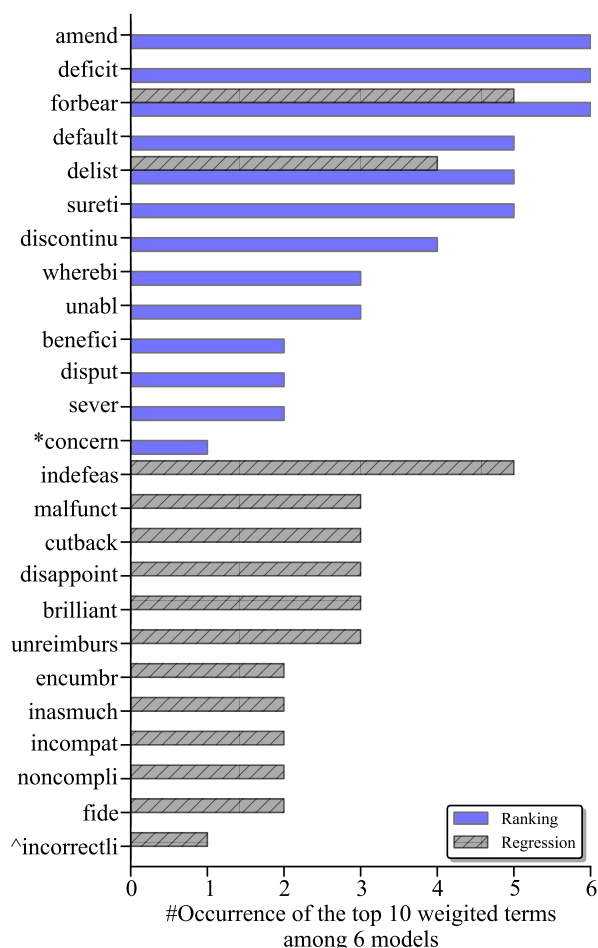


Figure 1: Number of Occurrences of the Top 10 Weighted Terms Learned via the Ranking and Regression Tasks. The notation \* denotes that except the term “concern” there are other terms that occur only one time among 6 ranking models, which are listed as follows: *breach, profit, violat, regain, uncomplet, accid, abl, integr, doubt, grantor*; similarly, for the notation ^, the terms are: *incorrectli, fault, nondisclosur, misus, breakag, defalc, excit, unclear, sentenc, overdu, omit, inforc, irrevoc, unencumb, further, variant, precipit, libel, loss*.

and text information might be a more reasonable way than the regression models.

## 4.2 Financial Sentiment Terms Analysis

As shown in Section 4.1, the ranking models can obtain more consistent results than the regression ones. Therefore, in the following discussions, we conduct some analyses on the words learned from the ranking models.

Figure 2 plots the words learned from our ranking models. In the figure, the single-outline circle denotes that only sentiment words are used as the training data; the double-outline circle denotes that all words in the original texts are considered when training. Moreover, the color filled in a circle with a term denotes which the sentiment word lists the

term belongs to; the circle with 2-mixed colors indicates the term belongs to two word lists. Note that the circle area is proportional to the average weight of each term.

In Figure 2, the top 5 average weighted words for the results of each kind of training data are marked by numbers from 1 to 5. For the case of training on sentiment words only (SEN), the top 5 average weighted words are *amend, deficit, forbear, delist, default*, whereas those under case ORG are *ceg, nasdaq, gnb, coven, forbear*; only one word *forbear* overlaps. An interesting finding is that when the models are trained on the original texts, some less informative terms like *ceg* (a company name, Co-Energy Group), *nasdaq* (an American stock exchange), *gnb* (a company name, GNB Technologies), are highly ranked; however, the relation is weak between these words and financial risk. In contrast, as only sentiment words are used for training, it is more reasonable that the terms are highly related to financial risk. In addition, since the terms in the figure have been stemmed, one term may correspond to one or more words. We also list the original words from the sentiment lexicon for each top 5 average weighted sentiment term in Figure 2. For example, the top 1 weighted term “*amend*” will have the list containing the words “*amend,*” “*amendable,*” “*amendatory,*” and so on.

Below we provide some original descriptions from 10-K reports that contain the top 2 weighted sentiment words in Figure 2. Note that the term with a higher weight is associated with higher financial risk. First, the term “*amend*” from the Fin-Lit list is considered. One piece of paragraph quoted from the original report is listed as follows:

(from AGO, 2006 Form 10-K)

On March 22, 2005, we *amended* the term loan agreements to, among other reasons, lower the borrowing rate by 25 basis points from LIBOR plus 2.00% to LIBOR plus 1.75%.

In finance, the *amend* usually means “to change by some formal processes.” This top-ranked term indicates that companies amending their policies frequently are associated with relative high risk.

We then discuss the term “*deficit*” from the Fin-Neg list, which means an excess of liabilities over assets, of losses over profits, or of expenditure over income in finance. Therefore, it is natural to say that a company associated with higher deficit might

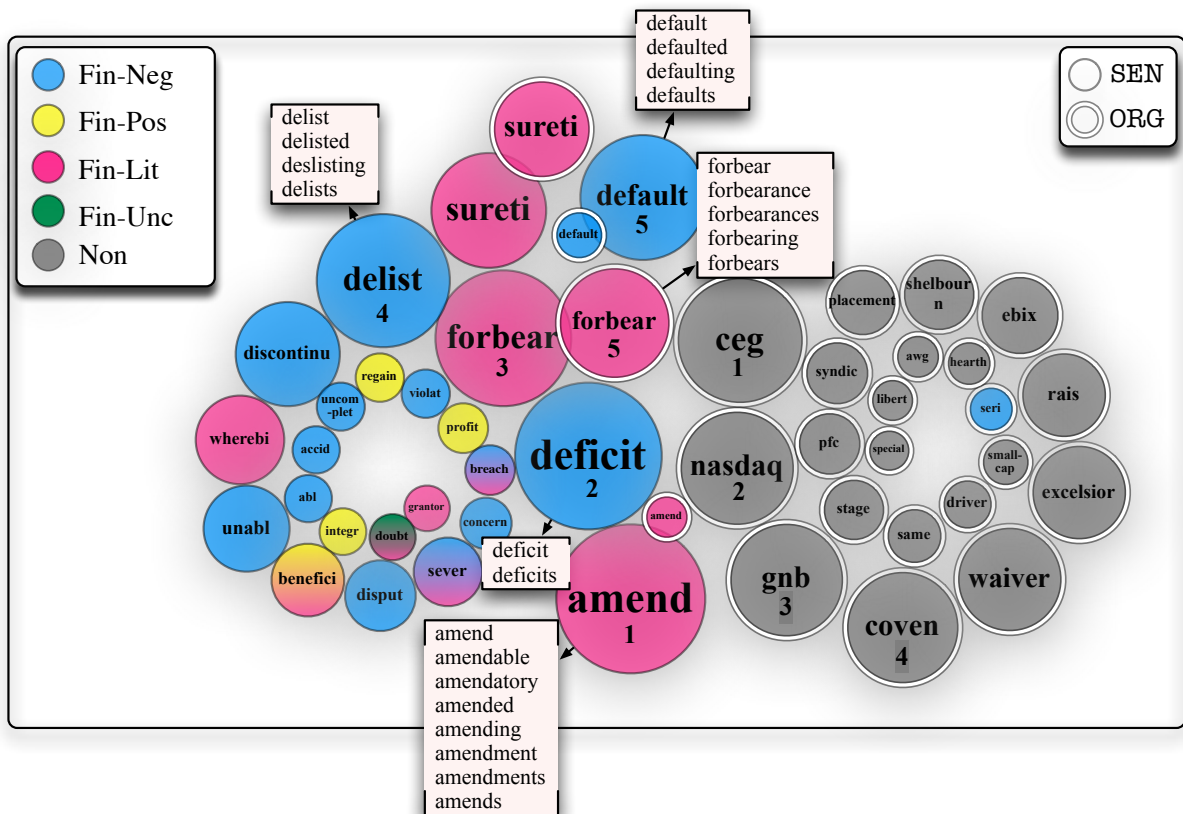


Figure 2: Highly-Weighted Terms Learned from the 6 Ranking Models of Using Original Texts (ORG) and Only Sentiment Words (SEN). The color filled in a circle with a term denotes which the sentiment word lists the word belongs to; the circle with 2-mixed colors indicates the term belongs to two word lists. The single-outline circle denotes that only sentiment words from the 6 dictionaries (see Table 2) are used as the training data; the double-outline circle denotes that the original texts are considered when training. Top 5 terms for the results of each kind of training data are marked by numbers from 1 to 5; the original words from the sentiment lexicon for each top 5 average weighted sentiment terms are also provided.

have higher risk. One piece of paragraph quoted from the original report is listed as follows:

(from AXS-One Inc., 2006 Form 10-K)  
 At December 31, 2005, we had cash and cash equivalents of \$3.6 million and a working capital *deficit* of \$3.6 million which included \$8.2 million of deferred revenue. The increase of the working capital *deficit* from \$3.3 million at December 31, 2004 is primarily the result of a decrease in cash and decreased accounts receivable offset partially by a decrease in deferred revenue.

## 5 Conclusions and Future Work

This paper identifies the importance of sentiment words in financial reports associated with financial risk. With the usage of a finance-specific sentiment lexicon, regression and ranking techniques are applied to analyze the relations between the sentiment words and financial risk. The experimental results

show that, based on the bag-of-words model, the models trained on sentiment words only can result in comparable performance to those on origin texts, which attests the importance of the financial sentiment words on risk prediction. In addition, the learned models also suggest strong correlations between financial sentiment words in financial reports and the risk of companies. As a result, these findings provide us more insight and understanding into the impact of financial sentiment words on companies' future risk. There are several future work, such as how to use even further information (i.e., syntactic information) for analysis, and how to conduct more fine-grained analysis.

## Acknowledgments

This research was partially supported by the National Science Council of Taiwan under the grants NSC 100-2218-E-133-001-MY2, 101-2221-E-004-017, 102-2221-E-004-006, and 102-2221-E-133-001-MY3.

## References

- H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. 1997. Support vector regression machines. *Advances in neural information processing systems*, pages 155–161.
- Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.
- Diego Garca. 2013. Sentiment during recessions. *The Journal of Finance (Online Published)*.
- Thorsten Joachims. 2006. Training linear svms in linear time. In *KDD '06*, pages 217–226.
- M.G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- S. Kogan, D. Levin, B.R. Routledge, J.S. Sagi, and N.A. Smith. 2009. Predicting risk from financial reports with regression. In *NAACL '09*, pages 272–280.
- Tim Loughran and Bill McDonald. 2011. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65.
- Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. In *NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34.
- J. L. Myers and A. D. Well. 2003. *Research design and statistical analysis*.
- Ramanathan Narayanan, Bing Liu, and Alok Choudhary. 2009. Sentiment analysis of conditional sentences. In *EMNLP '09*, pages 180–189.
- S McKay Price, James S Doran, David R Peterson, and Barbara A Bliss. 2012. Earnings conference calls and stock returns: The incremental informativeness of textual tone. *Journal of Banking & Finance*, 36(4):992–1011.
- B. Schölkopf and A.J. Smola. 2001. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*.
- Ming-Feng Tsai and Chuan-Ju Wang. 2013. Risk ranking from financial reports. In *Advances in Information Retrieval*, pages 804–807.
- R.S. Tsay. 2005. *Analysis of financial time series*.

# Sense disambiguation: from natural language words to mathematical terms

Minh-Quoc Nghiem,<sup>1</sup> Giovanni Yoko Kristianto,<sup>2</sup> Goran Topić,<sup>3</sup> Akiko Aizawa<sup>2,3</sup>

<sup>1</sup> The Graduate University for Advanced Studies, Tokyo, Japan

<sup>2</sup> The University of Tokyo, Tokyo, Japan

<sup>3</sup> National Institute of Informatics, Tokyo, Japan

{nqminh, giovanni, goran\_topic, aizawa}@nii.ac.jp

## Abstract

This paper addresses the open problem of mathematical term sense disambiguation. We introduce a method that uses a MathML parallel markup corpus to generate relevant training and testing datasets. Based on the dataset generated, we use Support Vector Machine classifier to disambiguate the sense of mathematical terms. Experimental results indicate we can generate such data automatically and with reasonable accuracy.

## 1 Introduction

Word-sense disambiguation (WSD) refers to the process of identifying the correct sense or meaning of a word in a sentence when the word has multiple meanings. WSD remains a difficult open problem in natural language processing. Current WSD systems are based on supervised, unsupervised, and knowledge-based approaches (Navigli, 2009). This paper focuses on the problem of disambiguating the sense of mathematical terms occurring within normal text, an aspect little discussed to date.

The problem of achieving automated understanding of mathematical expressions can be illustrated quite clearly. For instance, depending on context, the mathematical term  $\delta$  can be interpreted to refer to Kronecker Delta, Dirac Delta, Discrete Delta, or simply to a variable  $\delta$ . Another example is  $i$ , which can be interpreted to mean the imaginary constant, the index variable, or the bound variable of an operation. Other examples include  $\alpha$ ,  $\beta$ ,  $\sigma$ ,  $\phi$ ,  $\omega$ ,  $\Phi$ ,  $B$ ,  $H$ ,  $x$ ,  $y$ , *sim*. In many such cases, disambiguation can play a crucial role in the automated understanding, translation, and calculation of mathematical expressions.

One major issue in early research on machine understanding of mathematical terms found in text was the lack of evaluation datasets. A previous study (Wolska et al., 2011) was based on a small evaluation set of 200 mathematical expressions annotated by experts. Clearly, large samples of sense-tagged data would require significant human annotation and labor. Fortunately, then, Ide et al. (2002) showed that sense distinctions derived from cross-lingual information are at least as reliable as those made by human annotators. The novel research described in our paper presents a fully automated method for generating large samples of mathematical terms with sense-tagged data.

As part of the effort described here to address mathematical term sense disambiguation (MTSD), we first propose a method that uses a MathML parallel markup corpus to generate training and testing datasets. Second, we propose heuristics that improve alignment results for the parallel markup corpus. Third, we present a classification-based approach to the MTSD problem. To the best of our knowledge, this study is the first to make use of parallel corpora to address MTSD.

The rest of this paper is organized as follows: Sections 2 and 3 provide a brief overview of the background and related work; Section 4 presents our methods; Section 5 describes the experimental setup and results; Section 6 concludes the paper and points to directions for future research.

## 2 Background

Web pages and documents represent mathematical expressions in many formats: images,  $\text{\TeX}$ , MathML (Ausbrooks et al., 2010), OpenMath (Buswell et al., 2004), OMDoc (Kohlhase, 2006), or the ISO/IEC standard Office Open XML (Miller et al., 2009). This paper uses MathML markup, a format recommended by the W3C Math Working Group, as a standard for rep-

representing mathematical formulas. MathML uses presentation markup to capture notational structures and content markup to capture mathematical structures and mathematical meaning. MathML parallel markup provides both forms of markup for the same mathematical expression. Figure 1 shows the MathML presentation and content markup for the expression  $\arctan(0) = 0$ <sup>1</sup>.

#### Presentation MathML

```
<mrow>
  <mrow>
    <msup>
      <mi>tan</mi>
      <mrow>
        <mo>-</mo>
        <mn>1</mn>
      </mrow>
    </msup>
    <mo>(</mo>
    <mn>0</mn>
    <mo>)</mo>
  </mrow>
  <mo>=</mo>
  <mn>0</mn>
</mrow>
```

#### ContentMathML

```
<apply>
  <eq/>
  <apply>
    <arctan/>
    <cn>0</cn>
  </apply>
  <cn>0</cn>
</apply>
```

Figure 1: MathML presentation and content markup for the expression  $\arctan(0) = 0$

Natural language sentences and presentation mathematical expressions have several key similarities and differences. A token element in a mathematical expression can be regarded as a word in a sentence. In presentation markup, token elements are divided into four main types: identifiers ( $\langle \text{mi} \rangle x \langle \text{mi} \rangle$ ), operators ( $\langle \text{mo} \rangle + \langle \text{mo} \rangle$ ), numbers ( $\langle \text{mn} \rangle 2 \langle \text{mn} \rangle$ ), and text ( $\langle \text{mtext} \rangle \text{non zero} \langle \text{mtext} \rangle$ ). A sentence may contain certain layout elements, such as subscripts or superscripts, while a mathematical expression may contain numerous layout elements, such as  $\langle \text{mrow} \rangle$ ,  $\langle \text{msup} \rangle$ ,  $\langle \text{munderover} \rangle$ , and  $\langle \text{mfrac} \rangle$ . As noted by Ausbrooks et al. (2010), mathematical notation, while more rigorous than natural language, is ambiguous and context-dependent.

<sup>1</sup><http://functions.wolfram.com/01.14.03.0001.01>

### 3 Related Work

Several studies have shown encouraging results for WSD based on parallel corpora (Diab and Resnik, 2002; Tufiş et al., 2004; Chan and Ng, 2005; Carpuat and Wu, 2007; Padó and Lapata, 2009; Lefever and Hoste, 2010; Lefever et al., 2011). Ide et al. (2002) used translation equivalents derived from parallel aligned corpora to determine sense distinctions applicable to automatic sense-tagging. They evaluated their work using a subset of 33 nouns covering a range of occurrence frequencies and degrees of ambiguity (Ide et al., 2001), with results indicating no significant difference in agreement rates for the algorithm and for human annotators. The main limitation of this study is its dependence on aligned corpora, which are not easily obtainable.

Wolska et al. (Wolska and Grigore, 2010; Wolska et al., 2011) presented a knowledge-poor method for identifying the denotation of simple symbolic expressions in mathematical discourse. Based on statistical co-occurrence measures, the system sorted a simple symbolic expression under one of seven predefined concepts. Here, the authors found that lexical information from the linguistic context immediately surrounding the expression improved results. This approach achieves 66% agreement with the gold standard of manual annotation by experts. From our perspective, the predefined concepts are closely related to syntactic function, not the semantics of the terms.

### 4 Our Approach

#### 4.1 Generating the Datasets

We compiled our MTSD data using parallel MathML markup expressions gathered from the Web. First, using a set of heuristic rules, we pre-processed the parallel MathML markup expressions. We then used the GIZA++ toolkit to obtain node-to-node aligned data. Based on the node-to-node aligned data, we created subtree-to-subtree aligned data. Finally, we extracted ambiguous terms from the subtree-to-subtree aligned data to obtain data for MTSD. Figure 2 gives the steps taken to generate the data.

A crucial step in generating MTSD data is achieving alignment between the Presentation side and the Content side of the expressions. Given a set of several MathML parallel markup expressions, we used the automated word alignment



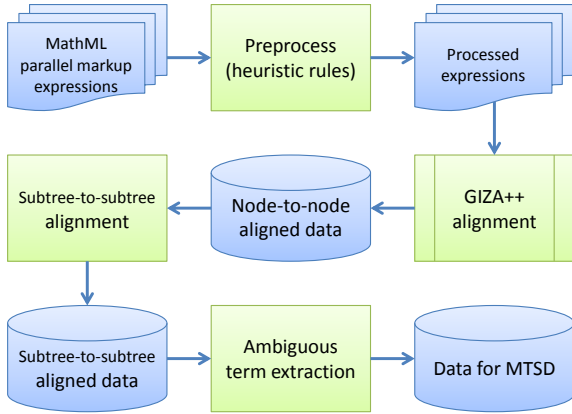


Figure 2: Steps for generating the data for MTSD.

GIZA++ (Och and Ney, 2003) to obtain alignment between the Presentation terms and Content terms. Developed to train word-based translation models, the GIZA++ toolkit is not directly applicable to a tree-based corpus. One common solution is to convert the tree into a sentence by extracting the leaf nodes of the tree and to form a sequence (Sun et al., 2010). While this approach works well for natural language text, it is less effective with mathematical expressions, since the intermediate nodes of these expressions contain layout information.

Before using GIZA++, to enhance alignment precision, we apply two heuristic rules to the presentation tree based on information on its structure. The first heuristic rule converts the intermediate layout nodes (except `msrow`) to leaves on the tree by moving them to the position of their first child. When moving an intermediate layout node, we create a temporary (‘temp’) node to replace the moved node and to keep the other child nodes intact. Unnecessary parentheses, which indicate that the expressions in the parentheses belong together, are also removed. Figure 3 illustrates an example of this heuristic. In this example, we moved the `msup` node to a leaf of the tree and removed a pair of parentheses, `<mo> (</mo>` and `<mo> ) </mo>`, near `<mn>0</mn>` node.

The second heuristic rule moves operator (`mo`) nodes to the beginning of the subtree if that subtree contains operator nodes. This rule reduces cross alignments, since most notations in content MathML are prefix notations and placed in leaf nodes. In Figure 3, the `<mo>=</mo>` node is moved to the first position of the tree. The `<mo>-</mo>` node is not moved because it is already the first child of its parent node. This figure also shows alignment results for GIZA++ before

and after applying heuristic rules for the expression  $\arctan(0)=0$ .

To extract more complex mathematical terms, we expand the node-to-node alignments to subtree-to-subtree alignments. In this study, we expanded the subtree alignment only to the parent of the `mi` nodes. The criteria used here to achieve subtree aligned pair are similar to that used by Tinsley et al. (2007). First, a node can be linked only once. Second, descendants of a presentation node can link only to descendants of its content counterpart. Third, ancestors of a presentation node can link only to ancestors of its content counterpart (a node counts as its own ancestor).

If one presentation node links to more than one content node, we keep only the link with the highest alignment score, as given by Equation 1. The number of alignments between the presentation tree  $tree_P$  and the content tree  $tree_C$  is the sum of (1) the number of alignments from the leaf children of  $tree_P$  to the leaf children of  $tree_C$  and (2) the number of alignments from the leaf children of  $tree_P$  to the leaf children of  $tree_C$ . For more accurate results, we removed node-to-node alignments if alignment probabilities fell below a certain threshold (0.2). In Equation 1,  $P_{child}$  and  $C_{child}$ , respectively, refer to the child nodes of  $tree_P$  and  $tree_C$ . The blue lines in Figure 3 represent the expanded alignments between subtrees.

$$score(tree_P, tree_C) = \frac{\# \text{alignments}}{\# P_{child} + \# C_{child}} \quad (1)$$

Based on the alignment results, we extracted pairs of presentation mathematical terms and their associated content terms. A mutually aligned presentation subtree and content subtree form a pair. This paper will consider only mathematical terms containing `mi` (e.g.  $\tan^{-1}$ ,  $A_i$ ,  $A_i(0)$ ,  $\Gamma$ ,  $\Gamma(\frac{2}{3})$ ). Only terms associated with ambiguous mapping are retained to generate training and testing data.

## 4.2 Disambiguating Mathematical Terms

We created a labeled training set, then used Support Vector Machines (SVM) to learn a classifier from this labeled data. Assume that a presentation term  $e$  has  $n$  ways of translating to content MathML term. Then, for each mathematical expression, we create one positive instance by combining  $e$  and its correct translation. We also create  $n-1$  negative instances by combining  $e$  and its incorrect translations. We will assign each instance



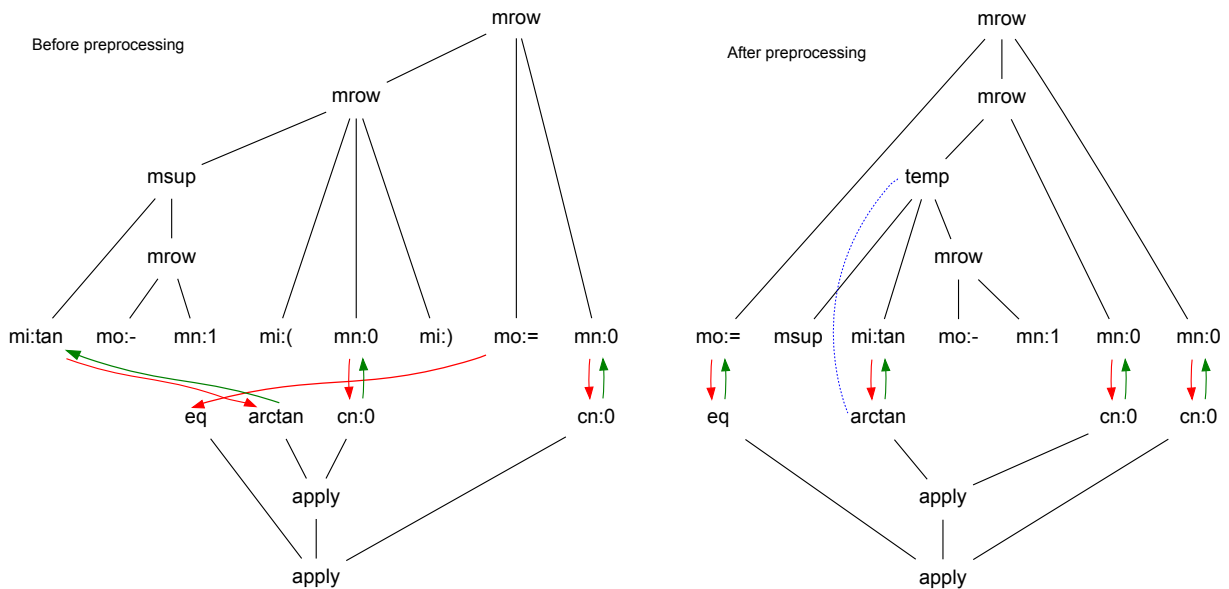


Figure 3: Example of alignment results for GIZA++ before and after applying the heuristic rules for the expression  $\arctan(0)=0$ . Red lines represent alignments from presentation nodes to content nodes; green lines represent alignments from content nodes to presentation nodes; blue lines represent expanded alignments between subtrees.

to one of two classes, depending on the candidate translation: The class is ‘true’ if the content term is the correct translation of the presentation term; otherwise, the class is ‘false.’

We can divide the features used in SVM disambiguation into two main groups: presentation MathML and text features. Presentation MathML features are extracted from the presentation MathML markup of the mathematical expressions. Mathematical compendium websites often group expressions into several categories. The only text feature we use here is the name of the category to which a mathematical expression belongs. Table 1 shows the features we used for classification.

Table 1: Features used for classification

Feature	Description
Only child	Is it the only child of its parent
Preceded by mo	Is it preceded by an $m_o$ node
Followed by mo	Is it followed by an $m_o$ node
mo’s name	The name of the followed $m_o$
Parent’s name	The name of its parent node
Node name	The name of the node
Identifier’s name	The name of the first $m_i$ child
Category	Relation between category name & candidate translation

Our experiment involved seven presentation MathML features. The first determines whether the term is the only child of its parent. The next three features encode the relationship between the term and the surrounding  $m_o$  elements. The last three features represent the parent’s name, the term’s own name, and the first  $m_i$  child’s name. Since mathematical terms differ from natural language words, the features differ as well.

## 5 Evaluation

### 5.1 Evaluation Setup

For these experiments, we collected parallel MathML markup expressions from the Wolfram Functions Site<sup>2</sup> (WFS), the world’s largest collection of formulas and graphics related to mathematical functions. All mathematical expressions on WFS are available in MathML parallel markup. For simplicity, we excluded long expressions containing more than 30 leaf nodes. We collected a total of 20,314 mathematical expressions.

### 5.2 Evaluation Results

We began by investigating the quality of the generated MTSD data. Using WFS data, we generated 2,925 different mathematical terms. There are 390 distinct ambiguous terms and 2,535 distinct

<sup>2</sup><http://functions.wolfram.com/>

unambiguous terms. Of the ambiguous terms, 90 distinct terms are single  $mi$  elements. There are 67,987 instances contain all the ambiguous terms in our data. Table 2 shows the generated data.

Table 2: Generated data

Type	Distinct term
Ambiguous $mi$ terms	90
Other ambiguous terms	300
Unambiguous terms	2,535

The table shows that only 14% of the extracted mathematical terms are ambiguous. One possible explanation: In WFS data, people tend to use one meaning for a fixed notation. Another: The system depends on the quality of the alignment output. The aligner may ignore an alignment if the probability of the alignment is low. This also causes errors in sense extraction if a sub-tree is aligned with a single term but the links are not fully connected: for example,  $\tan^{-1}$  (Presentation) and  $\arctan$  (Content).

Within the scope of this paper, we focused on the single  $mi$  element terms. (The same method can be expanded to encompass additional ambiguous terms.) We manually verified these single  $mi$  element terms to assess the quality of the generated MTSD data. Of 247 extracted senses, 197 were correct, an accuracy rate of 79.76% for the generated data. Each  $mi$  element term has an average of 2.74 senses. The term with the most senses was  $\langle mi \rangle C \langle /mi \rangle$ , which had six senses: Catalan, CatalanNumber, C, GegenbauerC, Cyclo-tomic, and FresnelC.

Next, we set up an experiment using libSVM<sup>3</sup> in the Weka toolkit (Hall et al., 2009) to examine sense disambiguation results for each presentation MathML term. The data we used contained the 90 distinct ambiguous  $mi$  terms. In this evaluation, we compared the results for systems using different training data: automatically extracted data and manually verified data. We also compared the results of our approach to the ‘most frequent’ method, which chooses the interpretation of highest probability. Since in the real world not every mathematical expression is associated with its category name, we also set up another experiment to assess the performance of our approach with and without the ‘category’ feature.

We built two models using nine-tenths of the au-

<sup>3</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

tomatically extracted data and nine-tenths of the manually verified data. Both systems set aside one-tenth of the verified data for testing. Classification accuracies were computed over the set of binary decisions. We used the default libSVM parameters. Table 3 gives the disambiguation accuracy for ambiguous  $mi$  terms.

Table 3: Sense disambiguation accuracy for ambiguous  $mi$  terms

Method	Extracted data	Verified data
All feature	<b>91.40</b>	<b>93.94</b>
Without ‘category’ feature	91.22	92.41
Most frequent	85.01	89.76

The results in Table 3 indicate reasonable results for the automatically extracted data. We gained improvements ranging from 1.2 to 2.5 percent by building a model using manually verified data. The classifier with ‘category’ feature slightly outperformed the classifier without the ‘category’ feature. Overall, the results here were approximately 4 to 7 percent more accurate than for the ‘most frequent’ method. The explanation for the relatively high scores for the ‘most frequent’ method is that mathematical elements often have a preferred meaning.

The results suggest we can make direct use of automatically generated data when working on the MTSD problem. For mathematical expressions in MathML parallel markup, the generated data is good enough without manual checking. The results also show that the text feature-i.e., the category of the mathematical term-contributes to system performance. While this improvement is modest, it suggests that features aside from the mathematical term itself can be helpful. However, the system works well even without this feature.

## 6 Conclusion

This paper presents an approach to creating training data for the mathematical term sense disambiguation problem. Combining word-to-word alignment models and heuristic alignments, this approach shows that we can generate reasonably accurate MTSD data using parallel corpora. The data generated can then be used to train a classifier that allows automatic sense-tagging of mathematical expressions.

In contrast to natural language text, mathematical expressions require specific processing methods. More work needs to be done to establish the features best-suited to mathematical terms in a larger dataset. An extension of the model with more text and context features, in addition to the category feature, should prove interesting. Since the alignments between presentation and the content tree affect the generated data, improving alignment accuracy may boost system performance.

## Acknowledgments

This research was supported in part by the Japan Society for the Promotion of Science (JSPS) via a Grant-in-Aid for Scientific Research (Grant 24300062).

## References

- Ron Ausbrooks, Stephen Buswell, David Carlisle, Giorgi Chavchanidze, Stéphane Dalmas, Stan Devitt, Angel Diaz, Sam Dooley, Roger Hunter, Patrick Ion, et al. 2010. Mathematical markup language (MathML) version 3.0. W3C recommendation. *World Wide Web Consortium*.
- Stephen Buswell, Olga Caprotti, David P Carlisle, Michael C Dewar, Marc Gaetano, and Michael Kohlhase. 2004. The open math standard version 2.0. Technical report.
- Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *In The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 61–72.
- Yee Seng Chan and Hwee Tou Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, pages 1037–1042.
- Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 255–262.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.
- Nancy Ide, Toma Erjavec, and Dan Tufis. 2001. Automatic sense tagging using parallel corpora. In *In Proceedings of the 6th Natural Language Processing Pacific Rim Symposium*, pages 212–219.
- Nancy Ide, Tomaz Erjavec, and Dan Tufis. 2002. Sense discrimination with parallel corpora. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions-Volume 8*, pages 61–66.
- Michael Kohlhase. 2006. *An Open Markup Format for Mathematical Documents (Version 1.2)*. Lecture Notes in Artificial Intelligence, no. 4180. Springer Verlag, Heidelberg.
- Els Lefever and Véronique Hoste. 2010. Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10*, pages 15–20.
- Els Lefever, Véronique Hoste, and Martine De Cock. 2011. Parasense or how to use parallel corpora for word sense disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 317–322.
- Frederic P. Miller, Agnes F. Vandome, and John McBrewhster. 2009. *Office Open XML*. Alpha Press.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:1–69.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
- Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.
- Jun Sun, Min Zhang, and Chew Lim Tan. 2010. Exploring syntactic structural features for sub-tree alignment using bilingual tree kernels. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 306–315.
- John Tinsley, Ventsislav Zhechev, Mary Hearne, and Andy Way. 2007. Robust language pair-independent sub-tree alignment. In *In Proceedings of MT Summit XI -07*.
- Dan Tufis, Radu Ion, and Nancy Ide. 2004. Fine-grained word sense disambiguation based on parallel corpora, word alignment, word clustering and aligned wordnets. In *Proceedings of the 20th international conference on Computational Linguistics, COLING '04*.
- Magdalena Wolska and Mihai Grigore. 2010. Symbol declarations in mathematical writing. In *Towards a Digital Mathematics Library. Paris, France, July 7-8th, 2010*, pages 119–127.
- Magdalena Wolska, Mihai Grigore, and Michael Kohlhase. 2011. Using discourse context to interpret object-denoting mathematical expressions. In *Towards a Digital Mathematics Library. Bertinoro, Italy, July 20-21st, 2011*, pages 85–101.

# Adapting a State-of-the-art Anaphora Resolution System for Resource-poor Language

Utpal Kumar Sikdar<sup>1</sup>, Asif Ekbal<sup>1</sup>, Sriparna Saha<sup>1</sup>  
Olga Uryupina<sup>2</sup>, Massimo Poesio<sup>3</sup>

<sup>1</sup> Department of Computer Science and Engineering, IIT Patna, India,  
{utpal.sikdar, asif, sriparna}@iitp.ac.in

<sup>2</sup> University of Trento, Center for Mind/Brain Sciences, uryupina@unitn.it

<sup>3</sup> University of Essex, Language and Computation Group, poesio@essex.ac.uk

## Abstract

In this paper we present our work on adapting a state-of-the-art anaphora resolution system for a resource poor language, namely Bengali. Performance of any anaphoric resolver greatly depends on the quality of a high accurate mention detector. We develop a number of models for mention detection based on heuristics and machine learning. Our experiments show that, a language-dependent system can attain reasonably good performance when re-trained on a new language with a proper subset of features. The system yields the MUC recall, precision and F-measure values of 57.80%, 79.00% and 66.70%, respectively. Our experiments with other available scorers show the F-measure values of 59.47%, 49.83%, 31.81% and 70.82% for BCUB, CEAFM, CEAFE and BLANC, respectively.

## 1 Introduction

Anaphora/co-reference resolution is the task of identifying noun phrases that are used to refer to the same entity in a text. More precisely, let us assume that C1 and C2 are occurrences of two noun phrases (NPs) and both have a unique referent in the context in which they occur. Here C2 refers to C1 in the context. C1 is called antecedent and C2 is called anaphor. The noun phrases that may participate in co-reference relation are called mentions/markables. Various practical tasks require language technology; for example, information extraction and text summarization, can be performed more reliably if it is possible to automatically find parts of the text containing information about a given topic. Anaphoric information is also needed to solve several other such kinds of Natural Language Processing (NLP) problems.

Most of these works on supervised machine learning co-reference resolution have been developed for English (Soon et al., 2001; Ng and Cardie, 2002; Yang et al., 2003; Luo et al., 2004), due to the availability of large corpora such as ACE (Walker et al., 2006) and OntoNotes (Weischedel et al., 2008). BART, the Beautiful Anaphora Resolution Toolkit (Versley et al., 2008), (Ponzetto and Strube, 2006), (Poesio and Kabadjov, 2004), is the resultant of the project titled "Exploiting Lexical and Encyclopedic Resources For Entity Disambiguation" carried out at the Johns Hopkins Summer Workshop 2007. It can handle all the preprocessing tasks to perform automatic coreference resolution. A variety of machine learning approaches are used in BART; it mainly uses several machine learning toolkits, including WEKA, MaxEnt and Support Vector Machine (SVM).

Literature shows the significant amount of works in the area of anaphora resolution. But these (Pradhan et al., 2012; Ng, 2010; Poesio et al., 2010) are mainly in non-Indian languages. The works related to anaphora resolution in Indian languages are still at the nascent stage due to the following facts: Indian languages are resource constrained, i.e. corpus, annotated corpus, morphological analyzers, Part-of-Speech (PoS) taggers, named entity (NE) taggers, parsers etc. are not readily available. There have been few attempts for anaphora resolution in Indian languages. In 2011 a shared task on NLP Tools Contest on Anaphora Resolution in Indian Languages was organized in association with 9th International Conference on Natural Language Processing (ICON 2011)<sup>1</sup>. Four teams participated in this contest with the varying approaches (Chatterji et al., 2011; Dakwale and Sharma, 2011; Senapati and Garain, 2011; Ghosh et al., 2011).

In this paper we propose our work on anaphora

---

<sup>1</sup><http://ltrc.iit.ac.in/icon2011/contests.html>

resolution in Bengali, a resource poor language. We develop a number of models for mention detection. The mention detector developed with the supervised classifier, Conditional Random Field (Lafferty, 2001) performs best for the anaphora resolution. We identify and implement several features for mention detection as well for anaphora resolution. Detailed experiments were carried out on the development set to identify the most relevant set of features. Later on we use that particular configuration to report the final evaluation results on test data.

## 2 Brief Description of BART System Architecture

Our starting point of anaphora resolution system is the toolkit from (Versley et al., 2008), originally conceived as a modularized version of previous efforts from (Ponzetto and Strube, 2006; Poesio and Kabadjov, 2004; Versley, 2006; Broscheit et al., 2010). BART’s final aim is to bring together state-of-the-art approaches, including syntax-based and semantic features. The state-of-the-art anaphora resolution system, BART has five main components: preprocessing pipeline, mention factory, feature extraction module, decoder and encoder. In addition, an independent language plugin module handles all the language specific information and is accessible from any component. Each module can be accessed independently and thus adjusted to leverage the system’s performance on a particular language or domain. The preprocessing pipeline converts an input document into a set of linguistic layers, represented as separate XML files. The mention factory uses these layers to extract mentions and assign their basic properties (number, gender etc). The feature extraction module describes pairs of mentions  $M_i, M_j, i < j$  as a set of features. The decoder generates training examples through a process of sample selection and trains a binary classifier. Finally, the encoder generates testing examples through a (possibly distinct) process of sample selection, runs the classifier and partitions the mentions into coreference chains.

### 2.1 Models for Mention Detection

Robust mention detection is an essential component of anaphora resolution system in any language. BART supports different pipelines for mention detection. The choice of a pipeline de-

pends crucially on the availability of linguistic resources for a given language. The very first step of anaphora resolution process tries to identify the occurrence of mentions in the documents. In our original experimental datasets, three information were provided for each token: Part-of-Speech (PoS), phrase (or, chunk) and Named Entity (NE). We develop the following mention detection models:

1. **First Model:** In our first model we consider each noun phrase(NP) as a possible candidate of mention. Results of this model are shown in Table 1.
2. **Second Model:** In our second model we consider each Named Entity (NE) or pronoun (PRP) as a mention and its results are shown in Table 1.
3. **Third Model:** In the third model we take only person name or pronoun (PER/PRP) as a candidate of mention. Results in Table 1 show a little improvement in the performance for one document, however the performance for the other documents decrease.
4. **Fourth Model:** Here we use Conditional Random Field (CRF) based supervised classifier to detect mentions from a given text. We formulate the mention detection as a classification problem by assigning each token in the text a label, indicating whether it is a mention or not. Hence to learn a classifier at first we have to create a training data and have to derive the class values (either B-mention/I-mention/Others)<sup>2</sup> of all the tokens from the annotated data. We create a training set for mention detection based on the mentions present in the original training data. Evaluation results in Table 1 clearly show that this mention detection system is the best compared to the other three models. Details of this systems are mentioned in the following subsection.

### 2.2 Conditional Random Field(CRF) based Mention Detection System

To formulate the problem of mention detection using CRF(Lafferty, 2001), we consider the token

<sup>2</sup>Here B, I and O denote the beginning, internal and outside the entity mention

of a sentence as an element of the observation sequence and the corresponding class label as an element of its state sequence. We have used the C++ based CRF++ package <sup>3</sup>.

### 2.2.1 Features for Mention Detection

We train CRF with the following set of features.

1. **Context word:** The contextual information of a target entity plays a significant role to decide whether it is a potential candidate for being a mention (or markable). We use the preceding and following few tokens as the features.
2. **Word suffix and prefix:** Fixed length (say,  $n$ ) word suffixes and prefixes are used as the features for mention detection. These are the fixed length character strings stripped either from the rightmost (for suffix) or from the leftmost positions (for prefix) of the words. We included this feature with the observation that mentions, in general, share some common character sequences either at the beginning or at the end.
3. **Part-of-Speech (PoS) information:** PoS information of the token is effective for mention detection. We consider the PoS classes like NN (Common noun), NNP (Proper noun), PRP (Pronoun) etc. as important for mention detection.
4. **Chunk information:** Each mention belongs to the noun phrase and so its boundary identification is important. We use the chunk information provided with the datasets.
5. **Suffix list:** Variable length suffixes of a word are matched with the predefined list of useful suffixes which are helpful to detect person (e.g., *-bAbu*, *-der*, *-dI*, *-rA* etc.) and pronoun (e.g., *-tI*, *-ke*, *-der* etc.) names <sup>4</sup>. We prepared such lists from the training data. A binary valued feature is defined that fires if the current word contains any of these suffixes.
6. **Noun phrase preceding pronoun:** We observed that in many cases the pronoun appears immediately after the potential markable candidate. We define a binary-valued

<sup>3</sup><http://crfpp.sourceforge.net>

<sup>4</sup>Henceforth all the Bengali glosses are written in ITRANS notations available at <http://www.aczoom.com/itrans/>

Sr.	Mentions	DevData	precision	recall	F-measure
1	NP	Doc-1 Doc-2	26.08 25.76	99.16 99.62	41.30 40.93
2	NE /PRP	Doc-1 Doc-2	72.02 47.18	33.80 25.86	46.01 38.35
3	PER /PRP	Doc-1 Doc-2	82.47 92.47	13.13 25.86	22.65 40.42
4	CRF Classifier	Doc-1 Doc-2	88.17 91.77	41.62 70.50	56.55 79.74

Table 1: Results of different approaches for mention detection on development data

feature that is set to 1 for a pronoun (PRP) if it follows a noun phrase (NP).

7. **Named entity information:** The Named Entity (NE) class is used for identifying mentions. This is a very useful feature as the majority of the mentions belong to the different NE categories.
8. **Pronoun list:** We manually prepare a list of pronoun names (e.g., *jeMon*, *kAro*, *tAhole*, *onnyoKe* etc.) that do not participate in anaphora resolution. This discards pronouns that are not co-referent mentions.
9. **First word:** Noun phrases often appear at the beginning for the particular datasets that we have used, and these can most likely be the mentions. This feature is used to define whether the token is the first word in the sentence or not.
10. **Morphological features:** We extract morphological features from the shallow parser available at <sup>5</sup>. The features include *lemma* and number *information* (singular/plural) of the words.
11. **Fine-grained noun information:** The fine-grained information of nouns are extracted from the PoS tags. The feature checks whether the token is definite noun or demonstrative noun, and decides accordingly.

We present the results of mention detection module in Table 1. It shows that CRF based classifier attains the best performance. Inspired by these results, we identify mentions in test data using this CRF based classifier. We merge the development

<sup>5</sup>[http://lrc.iit.ac.in/showfile.php?filename=downloads/shallow\\_parser.php](http://lrc.iit.ac.in/showfile.php?filename=downloads/shallow_parser.php)

TestData	precision	recall	F-measure
Doc-1	81.32	73.70	77.32
Doc-2	81.61	73.76	77.49
Doc-3	93.67	51.99	66.87

Table 2: Results for mention detection on test data

data with the training data and create a new training dataset for CRF. Results on the test data are reported in Table 2.

### 3 Methods for Anaphora Resolution

In this work we extend BART to perform anaphora resolution for Bengali, a resource poor language. We perform systematic study to identify most suitable configuration of BART for anaphora resolution in Bengali. We identify and implement several features for this task. We design and evaluate our system using the Bengali datasets obtained from the NLP Tools Contests on Anaphora Resolution in Indian Languages, organized in ICON-2011<sup>6</sup>. The Bengali corpus contains three types of datasets- training, development and test.

#### 3.1 Preprocessing and Markable Extraction

For the anaphora resolution system, mentions are identified from the datasets based on the gold annotations. These are treated as the markables. Thereafter we convert the markables to the data format used by BART, namely MMAX2s standoff XML format.

#### 3.2 Features for anaphora resolution

We view coreference resolution as a binary classification problem. We use the learning framework proposed by (Soon et al., 2001) as a baseline. Each classification instance consists of two markables, i.e. an anaphor and its potential antecedent. Instances are modelled as feature vectors and are used to train a binary classifier. The classifier has to decide, given the features, whether the anaphor and the candidate antecedent are coreferent or not. To improve the performance we define some features specific to the language. Given BART’s flexible architecture, we explore the contribution of some features implemented in BART for co-reference resolution in Bengali. Given a potential antecedent  $RE_i$  and an anaphor  $RE_j$ , we compute the following features:

1. **String match:** This feature compares between the two mentions. The value of this feature is true if the candidate anaphor ( $RE_j$ ) and antecedent ( $RE_i$ ) have the same surface strings forms, otherwise false.
2. **Sentence distance:** A non-negative integer feature capturing the distance between anaphor and antecedent; if they are in the same sentence, then value of 0 is produced else if their sentence distance is 1 the value of 1 is produced.
3. **Markable distance:** This is also a non-negative integer feature that captures the distance in terms of the number of mentions between the two markables.
4. **First person pronoun:** This feature is defined based on the direct and indirect speech. For a given anaphor-antecedent pair ( $RE_j, RE_i$ ) a feature is set to high if  $RE_j$  is a first person pronoun found within a quotation and  $RE_i$  is a mention immediately preceding it within the same quote. If  $RE_i$  is outside the quote and appears either in the same sentence or in any of the previous three sentences and is not first person then the corresponding feature is also set to high. The feature also behaves in a similar way if the pair ( $RE_j, RE_i$ ) appears outside the quotation.
5. **Second person pronoun:** This feature checks whether the pair ( $RE_j, RE_i$ ) is in the same quote and fires the feature accordingly. It is true if  $RE_j$  is second person and  $RE_i$  is other than the first person. If  $RE_j$  is inside the quotation, and  $RE_i$  ends with the suffix "ke" and is outside the quote then the feature fires.
6. **Third person pronoun:** This feature checks whether the pair ( $RE_j, RE_i$ ) appears inside or outside the quotation. It feature fires if both the mentions either appear within or outside the quotation.
7. **Reflexive pronoun:** For a given pair ( $RE_j, RE_i$ ), this feature checks whether  $RE_j$  is a reflexive pronoun and fires accordingly. This means if any antecedent is immediately followed by a reflexive pronoun then the feature is true, otherwise false.

<sup>6</sup><http://ltrc.iit.ac.in/icon2011/contests.html>

8. **Number agreement:** If both anaphor( $RE_j$ ) and antecedent( $RE_i$ ) agree in their number information then the feature value is set to true, otherwise false. We extract this feature from the shallow parser available at <sup>7</sup>. The parser was not able to take longer sentences as inputs and so we had to pre-process the data before running the parser.
9. **Semantic class feature:** If both  $RE_j$  and  $RE_i$  agree in their semantic classes then this feature is set to true, otherwise false. In particular this feature checks whether the pair either belongs to *person class* or *organization class* or *location class*.
10. **Alias feature:** It checks whether  $RE_j$  is an alias of  $RE_i$  or not.
11. **Appositive feature:** If  $RE_j$  is in apposition to  $RE_i$  then the value of this feature is set to true, otherwise it is false.
12. **String kernel:** String kernel similarity is used to estimate the similarity between two strings based on the string subsequence kernel.
13. **Mention type:** Following (Soon et al., 2001), we have encoded mention types (name, nominal or pronoun) of the anaphor and the antecedent. In addition, we check whether the anaphor  $RE_j$  is a definite pronoun or demonstrative pronoun or merely a pronoun. We also check whether each of the entities in the mention pair denotes proper name.

### 3.3 Learning algorithm

In order to learn coreference decisions, we experiment with WEKA's (Witten and Frank, 2005) implementation of the C4.5 decision tree learning algorithm (Quinlan, 1993), with the above mentioned feature combinations. Instances are created following (Soon et al., 2001). We generate a positive training instance from each pair of adjacent coreferent markables. Negative instances are created by pairing the anaphor with any markable occurring between the anaphor and the antecedent. During testing, we perform a closest first clustering of instances deemed coreferent by the classifier. Each text is processed from left to right: each

<sup>7</sup>[http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow\\_parser.php](http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php)

Dataset	#sentences	#tokens
Training	881	10,504
Development	598	5,785
Test	572	6,985

Table 3: Statistics of the datasets

markable is paired with any preceding markable from right to left, until a pair labelled as coreferent is output, or the beginning of the document is reached.

### 3.4 Decoding

In the decoding step, the coreference chains are created by the best-first clustering. Each mention is compared with all of its previous mentions with a probability greater than a fixed threshold value, and is clustered with the highest probability. If none has probability greater than the threshold, the mention becomes a new cluster.

## 4 Evaluation

### 4.1 Dataset

For our experiments we use the data sets provided in the ICON NLP Tools Contest on Anaphora Resolution in Indian Languages. The datasets were taken from the Bengali literature (mostly from the short stories). All the datasets were provided with PoS, chunk and NE information. For training and development datasets, anaphoric annotations were provided by the organizers. However for test set there was no annotation available. In line with the annotations of training and development datasets, we manually annotated test dataset. Some statistics of the datasets are presented in Table 3.

### 4.2 Evaluation metrics and results

In order to evaluate the anaphora resolution system we use different scorers such as MUC (Vilain et al., 1995), B<sup>3</sup> (Bagga and Baldwin, 1998), CEAF (Luo, 2005) and BLANC (Recasens and Hovy, October 2011). We experiment with the different mention detectors for anaphora resolution. Table 4 shows the MUC recall, precision and F-measure values of the system trained using the training data and evaluated using the development data. Experiments were carried out on a high performance computing facility with the following configuration: Dell machine, 216 cores, 2.66 GHZ Intel Xeon processors, 4 GB RAM/core, and 10 TB storage.



Mentions	recall	precision	F-measure
NP	52.50	40.40	45.60
NE/PRP	45.20	69.40	54.80
PER/PRP	45.20	66.30	53.80
CRF Classifier	52.20	78.80	62.80

Table 4: Results with MUC scorer on development data

Scorers	recall	precision	F-measure
MUC	57.80	79.00	66.70
BCUB	51.02	71.27	59.47
CEAFM	49.83	49.83	49.83
CEAFE	48.88	23.58	31.81
BLANC	70.66	70.99	70.82

Table 5: Overall results on test data

Results of Table 4 reveal the fact that the proposed anaphora resolution system achieves the best performance when CRF based classifier is used for mention detection. Based on these results on development data, we evaluate the system for the test data using the mentions extracted by the CRF based machine learner. Results on the test data are reported in Table 5. Results show the F-measure values of 66.70%, 59.47%, 49.83%, 31.81% and 70.82% for MUC, BCUB, CEAFM, CEAFE and BLANC, respectively.

### 4.3 Discussion

We explore different models for mention detections. We observed that the mention detection performs best with the supervised machine learner, CRF. These system mentions are then used for the encoding and decoding modules in BART. Experimental results shown in Table 4 show that mention detection plays an important role in anaphora resolution. We implement the baseline model using a subset of the features reported in (Soon et al., 2001). These include number agreement, alias, string matching, semantic class agreement, sentence distance, appositive and several features (c.f. Section 3.2). This model showed the MUC recall, precision and F-measure values of 38.8%, 67.4% and 49.3%, respectively. This is clearly much less compared to our proposed model. Comparisons with the available related works (specific to the language) show that our proposed system achieves state-of-the-art accuracy.

## 5 Conclusion

We present an anaphora resolution system for Bengali, a resource-poor language based on BART, a state-of-the-art coreference resolution model originally developed for English. We explore many models for markable identification, and observed that a supervised CRF based classifier produces the best results. The main focus of this work is to build a machine learning based anaphora resolution system for a resource-poor Indian language. Our system attains the state-of-the-art accuracy level. Currently our focus is on developing methods for capturing the missing markables; and identifying more syntactic and semantic features. Future work will also concentrate on porting the systems to other Indian languages, e.g. Hindi and Telugu, as well as investigating the portability and usefulness of more syntactic, morphological and semantic information across different languages. We also aim to perform systematic feature selection for mention detection and anaphora resolution.

## 6 Acknowledgments

The research described in this paper has been partially supported by the European Community’s Seventh Framework Programme (FP7/2007-2013) under the grant #288024: LIMOSINE – Linguistically Motivated Semantic aggregation engiNes.

## References

- Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proc. of the LREC workshop on Linguistic Coreference*, pages 563–566, Granada.
- Samuel Broscheit, Massimo Poesio, Simone Paolo Ponzetto, Kepa Joseba Rodriguez, Lorenza Romano, Olga Uryupina, Yannick Versley, and Roberto Zanoli. 2010. BART: A multilingual anaphora resolution system. In *Proceedings of the 5th International Workshop on Semantic Evaluations (SemEval-2010)*, Uppsala, Sweden, 15–16 July 2010, pages 104–107.
- Sanjay Chatterji, Arnab Dhar, Biswanath Barik, Moumita PK, Sudeshna Sarkar, and Anupam Basu. 2011. Anaphora resolution for Bengali, Hindi, and Tamil using random tree algorithm in weka. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011*, <http://ltrc.iit.ac.in/icon2011/contests.html>.

- Praveen Dakwale and Himanshu Sharma. 2011. Anaphora resolution in Indian languages using hybrid approaches. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011*, <http://ltrc.iiit.ac.in/icon2011/contests.html>.
- Aniruddha Ghosh, Snehasis Neogi, Saikat Chakrabarty, and Sivaji Bandyopadhyay. 2011. Anaphora resolution in Bengali. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011*, <http://ltrc.iiit.ac.in/icon2011/contests.html>.
- John Lafferty. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann.
- Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, A Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of the ACL*, pages 135–142.
- Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proc. NAACL / EMNLP*, Vancouver.
- Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111.
- Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411.
- Massimo Poesio and Mijail A. Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *Proceeding of LREC*, pages 663–666.
- Massimo Poesio, Simone Paolo Ponzetto, and Yannick Versley. 2010. Computational models of anaphora resolution: A survey.
- Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 192–199, New York City, USA, June. Association for Computational Linguistics.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea, July. Association for Computational Linguistics.
- J. Ross Quinlan. 1993. *programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Marta Recasens and Eduard Hovy. October, 2011. BLANC: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(04):485–510.
- Apurbalal Senapati and Utpal Garain. 2011. Anaphora resolution system for Bengali by pronoun emitting approach. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011*, <http://ltrc.iiit.ac.in/icon2011/contests.html>.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544, December.
- Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. Bart: A modular toolkit for coreference resolution. In *HLT-Demonstrations '08 Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 9–12.
- Yannick Versley. 2006. A constraint-based approach to noun phrase coreference resolution in german newspaper text. In *Proceedings of Konferenz zur Verarbeitung Nat rlicher Sprache*, pages 143–150.
- Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proc. of the Sixth Message Understanding Conference*, pages 45–52.
- Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus, linguistic data consortium, philadelphia.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, and Ann Houston. 2008. Ontonotes release 2.0:ldc2008t04 philadelphia penn.: Linguistic data consortium.
- Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 176–183.

# Chinese Event Coreference Resolution: Understanding the State of the Art

Chen Chen and Vincent Ng

Human Language Technology Research Institute

University of Texas at Dallas

Richardson, TX 75083-0688

{yzcchen, vince}@hlt.utdallas.edu

## Abstract

Given the relatively small amount of work on event coreference resolution, our understanding of the task is arguably fairly limited. This makes it difficult to determine how to improve an event coreference resolver. We seek to gain a better understanding of the state of the art in event coreference resolution by performing the first publicly available analysis of a Chinese event coreference resolver.

## 1 Introduction

Event coreference resolution is the task of determining which event mentions in a text refer to the same real-world event. Compared to entity coreference (the task of determining which entity mentions in a text refer to the same real-world entity), there is much less work on event coreference.

Given the lesser amount of work on event coreference, our understanding of the task is arguably fairly limited. Specifically, while it is not surprising that the performance of an event coreference resolver depends heavily on the quality of the output of its preprocessing components, it is not clear *to what extent* the noise inherent in the output of each preprocessing component is limiting the performance of a resolver. Note that this issue is more serious for event coreference than for entity coreference: since an event coreference resolver lies towards the end of the information extraction pipeline, it has to rely on the noisy output produced by more upstream components than its entity counterpart. The lack of understanding of this issue makes it difficult to identify the components that need most attention. Moreover, when analyzing the errors made by a resolver, it is not clear which types of errors can be fixed by improving the resolution algorithm and which ones can be fixed by improving the preprocessing com-

ponents. This makes it difficult to precisely determine how to improve the resolution algorithm. Taken together, these two issues make it difficult to understand how an event coreference resolver should be improved.

Our goal in this paper is to better understand the state of the art in event coreference and provide directions for further work on this task by addressing the aforementioned issues, which are summarized by the following two questions. First, to what extent is the noise inherent in the output of *each* of its upstream components limiting its performance? Second, what are the major types of errors that are attributable to (and therefore can be fixed by improving) the resolution algorithm?

We address these two questions by presenting a systematic analysis of a state-of-the-art Chinese event coreference resolver. Our decision to focus on Chinese can be attributed in part to the lack of publicly available results on Chinese event coreference resolution. In particular, to our knowledge, almost all recent work on event coreference has reported results for English (e.g., Humphreys et al. (1997), Chen et al. (2009), Bejan and Harabagiu (2010), Chen et al. (2011), Lee et al. (2012)).<sup>1</sup> Hence, the results in the paper can serve as a baseline against which future work on Chinese event coreference can be compared.

It is worth mentioning that similar questions were faced by entity coreference researchers, who have reached a point where it is crucial to answer these questions in order to make further progress. For that reason, a number of recent research papers have focused on these questions via analyzing the inner workings of entity coreference resolvers (e.g., Stoyanov et al. (2009), Recasens and Hovy (2010), Chen and Ng (2012a)). On the other hand, no attempts have been made to address these questions in the context of event coreference.

<sup>1</sup>A notable exception is Zinmeister et al.'s (2012) work, which reports results for both German and English.

## 2 ACE Event Coreference

As mentioned in the introduction, event coreference is the task of determining which event mentions in a text refer to the same real-world event. The ACE 2005 event coreference task, which is the version of the event coreference task we focus on, requires that an event coreference resolver performs coreference only on event mentions belonging to one of the ACE event types.

More specifically, an ACE event mention has a *type* and a *subtype*. In ACE 2005, eight types are defined, which are further subcategorized into 33 subtypes. Not surprisingly, two event mentions that have different subtypes cannot be coreferent.

To better understand the ACE 2005 coreference task, consider the sentence in Figure 1, which is taken from the ACE 2005 corpus. This example contains three event mentions: 砍 (stabbed), 伤 (injured) and 行凶 (criminal). 砍 and 行凶 have type CONFLICT and subtype ATTACK, whereas 伤 has type LIFE and subtype INJURE. In this example, 砍 and 行凶 are coreferent because they refer to the same real-world event.

---

(张家荣)(昨天傍晚)在(路上)骑自行车运动时遭到了(两名歹徒)持(刀)[砍][伤]。(歹徒)的[行凶]动机可能和(张家荣)的问证有关联。

(Zhang Jiarong) was cycling on (the road) (yesterday evening) and was [injured] when (two men) [stabbed] (him) with (a knife). (The thugs)' [criminal] motivation may have something to do with (Zhang Jiarong)'s testimony in a criminal case.

---

Figure 1: An example. Event mentions are bracketed and entity mentions are parenthesized.

## 3 Six Upstream Components

Our event coreference resolver adopts a fairly standard ACE event coreference system architecture, relying on six components. As we will see, the first four components have a direct influence on event coreference, meaning that their output will be used to create features for use by the event coreference model. On the other hand, the last two components only have an indirect influence on event coreference through other components.

**Component 1: Event mention boundary identification and subtyping.** This component (1) provides the event mentions for event coreference resolution, and (2) labels each event mention with its event subtype. Since two event mentions with different subtypes cannot be coreferent, subtypes

can be used to create useful features for event coreference. To implement this component, we use our Chinese event extraction system (Chen and Ng, 2012c), which jointly learns these tasks.

**Component 2: Event mention attribute value computation.** This component takes as input a set of event mentions (provided by Component 1) and computes for each mention its attributes, including its POLARITY, MODALITY, GENERICITY and TENSE. Since two event mentions that differ with respect to any of these attributes cannot be coreferent, they can be used to create useful features for event coreference. Following Chen et al. (2009), we train a classifier to compute the value of each attribute of each event mention (see Chen et al. for details on the implementation of these classifiers).

**Component 3: Event argument and role classification.** This component takes as input a set of event mentions (provided by Component 1) and a set of candidate arguments (provided by Component 5). For each event mention *em*, it (1) identifies those candidate arguments that are the true arguments of *em* (e.g., the participants, time, and place of *em*), and then (2) assigns a *role* (e.g., VICTIM, PLACE, TIME-WITHIN) to each of its true arguments. Since two events involving different times, places, or participants cannot be coreferent, the arguments and their roles can be used to create useful features for event coreference. To implement this component, we use our Chinese event extraction system (Chen and Ng, 2012c), which jointly learns these two tasks.

**Component 4: Entity coreference resolution.** This component takes as input a set of entity mentions (provided by Component 5) and creates a coreference partition in which each cluster contains all and only those entity mentions that refer to the same real-world entity. Since two event mentions having coreferent arguments are likely to be coreferent, the output of this component can be used to create useful features for event coreference. To create a coreference partition from a set of entity mentions, we employ our Chinese entity coreference resolver (Chen and Ng, 2012b).

**Component 5: Entity mention boundary identification.** This component provides the candidate arguments and the entity mentions needed by the aforementioned components, so it only has an indirect influence on event coreference. Since candidate arguments can be entity mentions, time ex-

pressions, and value expressions<sup>2</sup>, we train one CRF (using CRF++<sup>3</sup>) to extract each of these three types of candidate arguments.

#### Component 6: Entity typing and subtyping.

This component takes a set of entity mentions (provided by Component 5) and determines the semantic type and subtype of each entity mention. Knowing the semantic type and subtype of an argument is helpful for classifying the role of event arguments. For example, we can assign the role VICTIM only to those arguments with entity type PERSON. To determine semantic type and subtype, we train two SVM multiclass classifiers using SVM<sup>multiclass</sup> (Tsochantaridis et al., 2004).

### 4 Chinese Event Coreference Resolver

Underlying our learning-based Chinese event coreference resolver is a mention-pair model (Soon et al., 2001) trained using the SVM<sup>light</sup> package (Joachims, 1999). Training instances are created as follows. For each anaphoric event mention  $em$ , we create one positive instance between  $em$  and its closest antecedent. To create negative instances, we pair  $em$  with each of its preceding event mentions that is not coreferent with it.

Each instance is represented using 32 features, which are modeled after a state-of-the-art English event coreference resolver (Chen and Ji, 2009; Chen et al., 2009) (see the Appendix for a detailed description of these features). After training, the resulting mention-pair model is used in combination with a closest-first single-link clustering algorithm to impose a coreference partition on the event mentions in a test text (Soon et al., 2001).

### 5 Empirical Analysis

Next, we address our first question: to what extent is the noise inherent in the output of *each* of the upstream components limiting a resolver's performance? To answer this question, we start with a resolver where all of its upstream components are assumed to be oracle components, and then replace each of them with its real (i.e., imperfect) version one after the other, as described below.

#### 5.1 Experimental Setup

For evaluation, we report five-fold cross-validation results over the 633 Chinese documents

<sup>2</sup>See the ACE 2005 task definition (<http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v2a.pdf>).

<sup>3</sup><http://crfpp.googlecode.com/svn/trunk/doc/index.html>

in the ACE 2005 training corpus. Results, expressed in terms of recall (R), precision (P), and F-measure (F), are obtained by applying three commonly-used coreference scoring programs, MUC (Vilain et al., 1995), B<sup>3</sup> (Bagga and Baldwin, 1998), and CEAF<sub>e</sub> (a.k.a.  $\phi_4$ -CEAF) (Luo, 2005), to the coreference partitions produced by our resolver after singleton event mentions are removed. Stanford's Chinese NLP and Speech Processing tool<sup>4</sup> is used for word segmentation, syntactic parsing, and dependency parsing.

#### 5.2 Results and Analysis

As mentioned above, we start with an event coreference resolver that assumes that all the six upstream components are error-free (see row 1 of Table 1 for its performance), and then replace each oracle component with its *system* (i.e., machine-learned) counterpart one after the other (rows 2-7 of Table 1). Therefore, the results in the last row of Table 1 correspond to the performance of an *end-to-end* event coreference resolver that relies solely on system components. Below, we discuss the impact of each component on coreference performance. Note that these components can be considered in a different order than what we show in this section. Here, we show one ordering in which the parent(s) of a component are considered *after* the component itself.

**Component 2 (Event mention attribute value computation).** First, we replace the oracle event mention attribute predictors with their system counterparts. Since there are four event mention attributes, namely, POLARITY, MODALITY, GENERICITY and TENSE, we trained four classifiers to predict the attribute values of an event mention. Our results suggest that each of these four classifiers is only marginally better than a simple majority baseline.<sup>5</sup> We then used the values predicted by these classifiers to compute features for the *test* instances for the event coreference resolver. Note that the features for the *training* instances for the resolver are computed based on gold rather than system event attribute values.

<sup>4</sup><http://nlp.stanford.edu/projects/chinese-nlp.shtml>

<sup>5</sup>Chen et al. (2009) trained classifiers to predict the attribute values of English event mentions. While their POLARITY, MODALITY, and GENERICITY classifiers perform only slightly better than a majority baseline, their TENSE classifier has reasonably good performance. This is perhaps not surprising: tense classification for English verbs is easier than for Chinese verbs since Chinese verb forms do not change according to tense.

	MUC			B <sup>3</sup>			CEAF <sub>e</sub>			Avg F
	R	P	F	R	P	F	R	P	F	
1. All oracle	80.4	70.0	74.8	88.4	79.7	83.8	57.3	66.8	61.7	73.4
2. + System event mention attribute values										
2a. system event mention attributes on test only	59.9	60.8	60.3	76.8	78.5	77.6	53.3	52.5	52.9	63.6
2b. no event mention attribute features	72.8	63.4	67.8	84.2	76.6	80.2	52.1	60.0	55.8	67.9
2c. system event mention attributes on train & test	72.5	64.5	68.3	83.8	77.4	80.5	53.1	59.9	56.3	68.3
3. + System event arguments and roles	71.2	61.2	65.8	83.9	74.9	79.1	49.9	58.0	53.6	66.2
4. + System entity coreference chains	61.6	58.5	60.0	79.0	75.7	77.3	49.1	51.5	50.3	62.5
5. + System entity types & subtypes	62.2	57.9	60.0	79.4	75.2	77.2	49.0	52.3	50.6	62.6
6. + System entity mention boundaries	63.3	57.4	60.2	80.2	74.4	77.2	48.2	52.8	50.4	62.6
7. + System entity mention boundaries & subtypes	37.4	36.7	37.1	72.8	71.1	71.9	40.6	41.1	40.8	49.9

Table 1: Results when oracle components are replaced with system components one after the other.

Given the poor performance of these attribute predictors, we hypothesize that coreference performance will drop considerably when the oracle attribute predictors are replaced with their system counterparts. Results of this experiment, shown in row 2a, are consistent with this hypothesis: in comparison to row 1, the Avg F-score (unweighted average of the MUC, B<sup>3</sup>, and CEAF<sub>e</sub> F-scores)<sup>6</sup> drops significantly by nearly 10%.<sup>7</sup> A natural question is: do these results represent an unnecessary amplification of the impact of event mention attributes on event coreference performance? To answer this question, we consider two alternative ways of employing the system event mention attribute values for event coreference resolution. One way is to simply discard all features created from these attributes when training and testing the event coreference resolver. Results of this experiment are shown in row 2b. Somewhat interestingly, we can see by comparing rows 2a and 2b that discarding these features does yield a less considerable drop in performance than employing them. Another way is to employ system attribute values to generate features for both the training and test instances for the event coreference resolver. Results of this experiment are shown in row 2c. In comparison to row 2b, we see that there is a slight, but insignificant, improvement in coreference performance. Consequently, we assume in the rest of our experiments that we employ system event mention attribute values on both the training set and the test set (i.e., the configuration in row 2c), since it seems to more accurately reflect the impact of event mention attributes on event coreference performance.

**Conclusion 1:** Improving the four event attribute classi-

<sup>6</sup>For ease of exposition, we follow the CoNLL 2011 and 2012 shared tasks and use Avg F when discussing results.

<sup>7</sup>All statistical significance test results reported in this paper are conducted using the paired *t*-test ( $p < 0.05$ ).

fiers could significantly improve event coreference.

**Component 3 (Event argument and role classification).** Next, we replace the oracle event argument and role classification component with its system counterpart. Results on the test set indicate that when gold event mentions and subtype, gold entity type and subtype, and gold entity mention boundaries are used, the F-scores of system argument classification and role classification are 76.9% and 68.2% respectively. Replacing the oracle component with this system counterpart, we see that the Avg coreference performance drops slightly, though still significantly, by 2.1%.

**Conclusion 2:** Event argument classification and role classification have a small, but significant, impact on event coreference performance.

**Component 4 (Entity coreference).** Next, we replace oracle entity coreference with system entity coreference. As noted before, event coreference directly depends on entity coreference. Our system entity coreference resolver achieves a MUC F-score of 78.0% when gold entity mentions are used. Comparing rows 3 and 4, we see that replacing oracle entity coreference with system entity coreference incurs nearly a 4% drop in coreference performance according to Avg F-score. These results suggest that employing a better entity coreference resolver can improve event coreference. Joint learning of event and entity coreference may help to improve both tasks.

**Conclusion 3:** Improving entity coreference could significantly improve event coreference.

**Component 6 (Entity typing and subtyping).** Next, we replace oracle entity typing and subtyping with its system counterpart. Recall that this component has only an indirect impact on event coreference but a direct impact on event argument and role classification, since the entity type and subtype of a mention are used to create features for training the event argument and role classifier. Our

system entity type and subtype classifiers achieve F-scores of 90.1% and 81.6%, respectively, when gold entity mentions are used. Using system rather than gold entity types and subtypes, the F-scores of the event argument classifier and the event role classifier drop by 2.8% and 4.3% respectively, but comparing rows 4 and 5, coreference performance does not drop.

**Conclusion 4:** Improving entity type and subtype classification is unlikely to improve event coreference.

**Component 5 (Entity mention boundary detection).** Next, we replace gold entity mention boundary detection with its system counterpart. The performance of our system mention boundary detection component is reasonably good: it achieves an F-score of 84.7%. Comparing rows 5 and 6, we see that replacing gold mention boundary detection with its system counterpart does not alter event coreference performance.

**Conclusion 5:** Improving entity mention boundary detection may not improve event coreference.

**Component 1 (Event mention boundary identification and subtyping).** Finally, we replace the oracle event mention boundary identification and subtyping component with its system counterpart. Our learned event mention boundary identifier achieves an F-score of 65.1%, while our event subtype classifier achieves an F-score of 61.30%. Comparing rows 6 and 7, we see that replacing oracle with system event mention boundary identification and subtyping causes Avg F-score to drop by 12.7%, even though the event extraction system we employ for event mention boundary identification and subtype classification is a state-of-the-art system. Furthermore, MUC recall decreases more abruptly than MUC precision, which suggests that the low recall of event mention boundary identification severely harms system performance.

**Conclusion 6:** Event mention boundary identification and subtyping is the upstream component that has the largest impact on event coreference. There is a lot of room for improving this component, especially its recall.

We conclude this section with two noteworthy points. First, the *cumulative* study we conducted in this section is just one possible way to examine the extent to which the noise inherent in the output of each of the upstream components limits a resolver's performance. Another way is to conduct an *ablation* study: we start with a resolver where all of its upstream components are assumed to be oracle components, and then replace exactly one

oracle upstream component with its real (i.e., imperfect) version in each ablation experiment. Since the conclusions that can be drawn from the ablation study and the cumulative study are similar, we will omit the description of the ablation experiments and their results for the sake of brevity.

Second, although we discuss the results in this section in terms of Avg F, it turns out that in these experiments Avg F exhibits the same performance trends as those of MUC, B<sup>3</sup>, and CEAF<sub>e</sub>. In particular, the significance test results that we obtained using Avg F remain unchanged when Avg F is replaced with any of the three scoring metrics.

## 6 Error Analysis

Next, we address our second question: what are the major types of errors that are attributable to (and therefore can be fixed by improving) the resolution algorithm? To answer this question, we perform a qualitative error analysis on the output produced by the resolver where all six upstream components are gold. This ensures that all the errors are attributable to the resolution algorithm.

### 6.1 Three Major Types of Precision Errors

**Lack of event timestamping.** Only those events that occur at the same time can be coreferent. We use the TENSE event attribute as a feature to enforce TENSE consistency, essentially employing TENSE as a very rough approximation of the timestamp of an event. Perhaps not surprisingly, many events having the same tense are not coreferent. For example, consider the following pair of sentences, where the *triggers* (i.e., the words/phrases corresponding to the event mentions) are enclosed in brackets.

去年三月间杨光南在上海首度 [被捕]

In last March, Yang Guangnan was [arrested] in Shanghai for the first time

杨光南在上海再度 [被捕]

Yang Guangnan was [arrested] again in Shanghai

It is fairly easy to see that the first *arrest* occurred before the second one and therefore the two event mentions should not be coreferent. However, our resolver incorrectly posits them as coreferent, since they have the same PERSON and PLACE arguments (i.e., 杨光南 (Yang Guangnan) and 上海 (Shanghai)) and the same tense (i.e., *past*). If we could assign a timestamp to each event, our resolver might fix this type of precision error.

**Incompatible triggers.** As coreference between arguments is a strong indicator that the corresponding event mentions are coreferent, our resolver tends to link two event mentions with coreferent arguments together even when the triggers are not semantically compatible. The following pair of sentences illustrates this type of error.

萨姆·努乔马 28 日乘专机抵达平壤开始对朝鲜进行正式友好 [访问]

On the 28th, Sam Nujoma arrived in Pyongyang by plane for an official goodwill [visit] to the DPRK

纳米比亚总统萨姆·努乔马 28 日乘专机 [抵达] 平壤  
Namibian President Sam Nujoma [arrived] in Pyongyang by plane on the 28th

As we can see, the triggers are 访问 (visit) and 抵达 (arrived). Since both their ARTIFACT arguments (i.e., 萨姆·努乔马 (Sam Nujoma)) and their VEHICLE arguments (i.e., 专机 (plane)) are coreferent, our resolver wrongly posits the two event mentions as coreferent, although the corresponding triggers, 访问 and 抵达, are semantically incompatible. If we had access to a semantic dictionary from which we can derive the fine-grained semantic type of these triggers, our resolver might fix this type of error.

**Incompatible important arguments.** Our resolver has the tendency to posit two *largely compatible* event mentions as coreferent, i.e., they have only a small number of incompatible arguments but many features that suggest that they are coreferent (e.g. same trigger word, some coreferent arguments). Consider the example below.

代表团 [访问] 了瑞典

The delegation [visited] Sweden

[访问] 丹麦期间, 中国基督教代表团举行记者招待会

During their [visit] in Denmark, the Chinese Christian delegation held a press conference

Note that the two event mentions have identical triggers 访问 (visited) and ARTIFACT arguments 代表团 (The delegation). Given these positive evidences, our resolver posits the event mentions as coreferent despite the fact that their DESTINATION arguments are incompatible: one is 瑞典 (Sweden) and the other is 丹麦 (Denmark). To fix this kind of error, we may employ human knowledge to mark each argument role of each event subtype as *important* or *unimportant*, and enforce the constraint that two event mentions cannot be coreferent if their arguments in an *important* argument role are incompatible. In our example, we would mark both ARTIFACT and DESTINA-

TION as important argument roles for the MEETING event subtype (i.e., the subtype for *visited*), meaning that we will disallow the two event mentions in the example to be coreferent unless both the ARTIFACT and DESTINATION arguments are compatible.

## 6.2 Two Major Types of Recall Errors

**Coreferent mentions with synonymous triggers.** Our resolver fails to link some event mentions that have synonymous but lexically different trigger words. Consider the following example.

犹太人针对阿拉伯人的 [暴力]

Jewish [violence] against the Arabs

[冲突] 双方

Two parties of [conflict]

While the event mentions corresponding to the two synonymous triggers, 暴力 (violence) and 冲突 (conflict), are coreferent, our resolver fails to identify them as coreferent because the triggers are lexically different. We could fix this type of error if we had access to a semantic dictionary that can suggest that *violence* and *conflict* have similar meaning.

**Coreferent mentions with compatible arguments.** Some arguments are not coreferent but compatible. Consider the following sentences.

南斯拉夫国家元首第一次对波黑进行这样的 [访问]

Yugoslavia's head of state [visited] Bosnia-Herzegovina for the first time

科什图尼察 [访问] 波黑首都萨拉热窝

Kostunica [visited] Sarajevo, the capital of Bosnia-Herzegovina

Here, the triggers are 访问 (visited). Our resolver does not posit these two event mentions as coreferent since their arguments are not coreferent. However, if we had the world knowledge to recognize 波黑 (Bosnia and Herzegovina) and 萨拉热窝 (Sarajevo) are compatible arguments, then our resolver might be able to discover the coreference link between the two event mentions.

## 7 Conclusion

We conducted the first empirical analysis of an ACE-style Chinese event coreference system, focusing on the questions of (1) the extent to which event coreference performance is affected by errors made by upstream components in the information extraction pipeline, and (2) the types of errors made by the resolution algorithm. We hope our analysis will help direct future research.



## Acknowledgments

We thank the four reviewers for their insightful comments. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142.

## Appendix: Event Coreference Features

The 32 features used by our event coreference resolver can be divided into five groups. Group  $i$  ( $1 \leq i \leq 4$ ) contains the features computed based on Component  $i$ 's output, and Group 5 contains the remaining features. For convenience, we use  $em_2$  to refer to an event mention to be resolved and  $em_1$  to refer to one of its candidate antecedents.

**Group 1 (4):** Whether  $em_1$  and  $em_2$  agree w.r.t. event type; whether they agree w.r.t. event subtype; the concatenation of their event types; the concatenation of their event subtypes.

**Group 2 (8):** The four event mention attributes of  $em_2$ ; whether  $em_1$  and  $em_2$  are compatible w.r.t. each of the event mention attributes.

**Group 3 (4):** The roles and number of the arguments that only appear in  $em_1$ ; the roles and number of the arguments that only appear in  $em_2$ .

**Group 4 (6):** The roles and number of arguments between  $em_1$  and  $em_2$  that have the same role and are also in the same entity coreference chain; the roles and number of arguments between  $em_1$  and  $em_2$  that have same role but are in different coreference chains; the roles and number of arguments between  $em_1$  and  $em_2$  that have different roles but are in the same coreference chain.

**Group 5 (10):** The sentence distance between  $em_1$  and  $em_2$ ; the number of event mentions intervening them; the number of words between them; whether they have the same trigger word; whether they are in a coordinating structure; whether they have same basic verb; whether they agree in number if they are nouns; whether they have incompatible modifiers if they are nouns; the concatenation of the part-of-speech tags of their heads; the concatenation of their trigger words.

## References

- Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proc. of the LREC Workshop on Linguistic Coreference*, page 563--566.
- Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proc. of the ACL*, pages 1412--1422.
- Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proc. of TextGraphs-4*, pages 54--57.
- Chen Chen and Vincent Ng. 2012a. Chinese noun phrase coreference resolution: Insights into the state of the art. In *Proc. of COLING 2012: Posters Volume*, pages 185--194.
- Chen Chen and Vincent Ng. 2012b. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Proc. of EMNLP-CoNLL: Shared Task*, pages 56--63.
- Chen Chen and Vincent Ng. 2012c. Joint modeling for Chinese event extraction with rich linguistic features. In *Proc. of COLING*, pages 529--544.
- Zheng Chen, Heng Ji, and Robert Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proc. of the RANLP Workshop on Events in Emerging Text Types*, pages 17--22.
- Bin Chen, Jian Su, Sinno Jialin Pan, and Chew Lim Tan. 2011. A unified event coreference resolution by integrating multiple resolvers. In *Proc. of IJCNLP*, pages 102--110.
- Kevin Humphreys, Robert Gaizauskas, and Saliha Azam. 1997. Event coreference for information extraction. In *Proc. of the ACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 75--81.
- Thorsten Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, pages 44--56. MIT Press.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proc. of EMNLP-CoNLL*, pages 489--500.
- Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proc. of HLT/EMNLP*, pages 25--32.
- Marta Recasens and Eduard Hovy. 2010. Coreference resolution across corpora: Languages, coding schemes, and preprocessing information. In *Proc. of the ACL*, pages 1423--1432.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521--544.
- Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proc. of ACL-IJCNLP*, pages 656--664.
- Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proc. of ICML*, pages 104--112.
- Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proc. of MUC-6*, pages 45--52.
- Heike Zinmeister, Stefanie Dipper, and Melanie Seiss. 2012. Abstract pronominal anaphors and label nouns in German and English: Selected case studies and quantitative investigations. *Translation: Corpora, Computation, Cognition*, 2(1):47--80.

# A Two-Step Named Entity Recognizer for Open-Domain Search Queries

**Andreas Eiselt**

Yahoo! Research Latin America  
Av. Blanco Encalada 2120,  
Santiago, Chile  
eiselt@yahoo-inc.com

**Alejandro Figueroa**

Yahoo! Research Latin America  
Av. Blanco Encalada 2120,  
Santiago, Chile  
afiguero@yahoo-inc.com

## Abstract

Named entity recognition in queries is the task of identifying sequences of terms in search queries that refer to a unique concept. This problem is catching increasing attention, since the lack of context in short queries makes this task difficult for full-text off-the-shelf named entity recognizers. In this paper, we propose to deal with this problem in a two-step fashion.

The first step classifies each query term as token or part of a named entity. The second step takes advantage of these binary labels for categorizing query terms into a pre-defined set of 28 named entity classes. Our results show that our two-step strategy is promising by outperforming a one-step traditional baseline by more than 10%.

## 1 Introduction

Search engines are key players in serving as interface between users and web resources. Hence, they started to take on the challenge of modelling user interests and enhance their search experience. This is one of the main drivers of replacing the classical document-keyword matching, a.k.a. bag-of-words approach, with user-oriented strategies. Specifically, these changes are geared towards improving the precision, contextualization, and personalization of the search results. To achieve this, it is vital to identify fundamental structures such as named entities (e.g., persons, locations and organizations) (Hu et al., 2009). Indeed, previous studies indicate that over 70% of all queries contain entities (Guo et al., 2009; Yin and Shah, 2010).

Search queries are on average composed of 2-3 words, yielding few context and breaking the grammatical rules of natural language (Guo et al., 2009; Du et al., 2010). Thus, named entity recognizers for relatively lengthy grammatically well-

formed documents perform poorly on the task of Named Entity Recognition in Queries (NERQ).

At heart, the contribution of this work is a novel supervised approach to NERQ, trained with a large set of manually tagged queries and consisting of two steps: 1) performs a binary classification, where each query term is tagged as token/entity depending on whether or not it is part of a named entity; and 2) takes advantage of these binary token/entity labels for categorizing each term within the query into one of a pre-defined set of classes.

## 2 Related Work

To the best of our knowledge, there have been a few previous research efforts attempting to recognize named entities in search queries. This problem is relatively new and it was first introduced by (Paşca, 2007). Their weakly supervised method starts with an input class represented by a set of seeds, which are used to induce typical query-contexts for the respective input category. Contexts are then used to acquire and select new candidate instances for the corresponding class.

In their pioneer work, (Guo et al., 2009) focused on queries that contain only one named entity belonging to four classes (i.e., movie, game, book and song). As for learning approach, they employed weakly supervised topic models using partially labeled seed named entities. These topic models were trained using query log data corresponding to 120 seed named entities (another 60 for testing) selected from three target web sites. Later, (Jain and Pennacchiotti, 2010) extended this approach to a completely unsupervised and class-independent method.

In another study, (Du et al., 2010) tackled the lack of context in short queries by interpreting query sequences in the same search session as extra contextual information. They capitalized on a collection of 6,000 sessions containing only queries targeted at the car model domain.

They trained Conditional Random Field (CRF) and topic models, showing that using search sessions improves the performance significantly. More recent, (Alasiry et al., 2012a; Alasiry et al., 2012b) determined named entity boundaries, combining grammar annotation, query segmentation, top ranked snippets from search engine results in conjunction with a web n-gram model.

In contrast, we do not profit from seed named entities nor web search results, but rather from a large manually annotated collection of about 80,000 open-domain queries. We consider search queries containing multiple named entities, and we do not benefit from search sessions. Furthermore, our approach performs two labelling steps instead of a straightforward one-step labelling. The first step checks if each query term is part of a named entity or not, while the second assigns each term to one out of a set of 29<sup>1</sup> classes by taking into account the outcome of the first step.

### 3 NERQ-2S

NERQ-2S is a two-step named entity recognizer for open-domain search queries. First, it differentiates named entity terms from other types of tokens (e.g., word and numbers) on the basis of a CRF<sup>2</sup> trained with manually annotated data. In the second step, NERQ-2S incorporates the output of this CRF into a new CRF as a feature. This second CRF assigns each term within the query to one out of 29 pre-defined categories. In essence, considering these automatically computed binary entity/token labels seeks to influence the second model so that the overall performance is improved.

Given the fact that binary entity/token tags are only used as additional contextual evidence by the second CRF, these labels can be reverted in the second step. NERQ-2S identifies 28 named entity classes that are prominent in search engine open-domain queries (see table 1). This set of categories was deliberately chosen as a means of enriching search results regarding general user interests, and thus aimed at providing a substantially better overall user experience. In particular, named entities are normally utilized for devising the lay-out and the content of the result page of a search engine.

<sup>1</sup>In actuality, we considered 29 classes: 28 regards named entities and one class for non-entity (token). For the sake of readability, from now on, we say indistinctly that the second step identifies 28 named entity classes or 29 classes.

<sup>2</sup>CRFsuite: <http://www.chokkan.org/software/crfsuite>

At both steps, NERQ-2S uses a CRF as classifier and a set of properties, which was determined separately for each classifier by executing a greedy feature selection algorithm (see next section). For both CRFs, this algorithm contemplated as candidates the 24 attributes explained in table 2. Additionally, in the case of the second CRF, this algorithm took into account the entity/token feature produced by the first CRF. Note that features in table 2 are well-known from other named entity recognition systems (Nadeau and Sekine, 2007).

## 4 Experiments

In all our experiments, we carried out a 10-fold cross-validation. As for **data-sets**, we benefited from a collection comprising 82,413 queries, which are composed of 242,723 terms<sup>3</sup>. These queries were randomly extracted from the query log of a commercial search engine, and they are exclusively in English. In order to annotate our query collection, these queries were first tokenized, and then each term was manually tagged by an editorial team using the schema adopted in (Tjong Kim Sang and De Meulder, 2003).

Attributes were selected by exploiting a **greedy** algorithm. This procedure starts with an empty bag of properties and after each iteration adds the one that performs the best. In order to determine this feature, this procedure tests each non-selected attribute together with all the properties in the bag. The algorithm stops when there is no non-selected feature that enhances the performance.

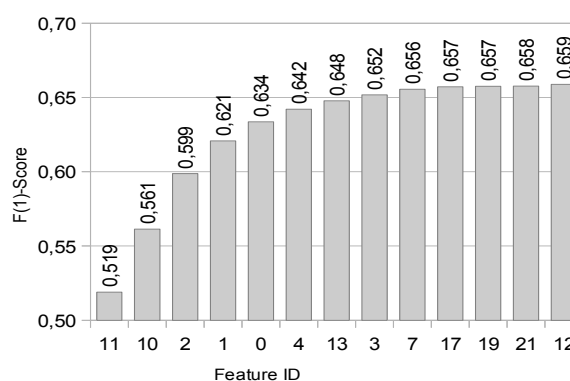


Figure 1: Attributes selected by the greedy algorithm and their respective contribution (baseline). See also table 2 for id-feature mappings.

As for a **baseline**, we used a traditional one-step approach grounded on CRF enriched with 13

<sup>3</sup>Due to privacy laws, query logs cannot be made public.

ID	Name	Example	ID	Name	Example
0	Airline Code	AA, LA, JJ	15	Food	Sushi, Bread, Dessert
1	Beverage	Cocktails, Beer	16	Food Ingredient	Honey, Avocado
2	Brand Name	Bacardi, Apple	17	Food Taste	Sweet, Cheesy
3	Business	Hotel, Newspaper	18	Horoscope Sign	Libra, Taurus
4	Cooking Method	Pressure Cooking	19	Measurement Name	Inches, Kilogram
5	Cuisine	Mexican, German	20	Media Title	Age of Empires 2
6	Currency Name	Dollar, Euros, Pesos	21	Occasion	Festival, Ceremony
7	Diet	Vegan, Fat free	22	Organization Name	Yahoo, Caf Soleil
8	Disease and Condition	Cancer, Diabetic	23	Person Name	Marry Poppins
9	Dish	Ratatouille, Tiramisu	24	Phone Number	3153423595
10	Domain	forbes.com, lan.com	25	Place Name	Chile, Berlin
11	Drink	Bloody Mary, Sangria	26	Product	Camera, Cell phone
12	Email Address	john.doe@example.com	27	Treatment	Steroids, Surgery
13	Event Name	Christmas, Super Bowl	28	<i>Token</i> (no NE-class)	how, to, image
14	File Name	msimn.exe, .htaccess			

Table 1: Named entity classes recognized by NERQ-2S.

out of our 24 features (see table 2), which were chosen by running our greedy feature selection algorithm. Figure 1 shows the order that these 13 features were chosen, and their respective impact on the performance. Regarding these results, it is worth highlighting the following findings:

1. The first feature selected by the greedy algorithm models each term by its non-numerical characters (id=11 in table 2). This attribute helps to correctly tag 80.42% of the terms when they are modified (numbers removed).
2. The third chosen feature considers the value of the following word, when tagging a term (id=2 in table 2). This attribute helps to correctly annotate 79.68%, 74.55% and 74.87% of tokens belonging to person, place and organization names, respectively.
3. Our figures also point out to the relevance of the three word features (id=0,1,2 in table 2). These features were selected in a row, boosting the performance from  $F(1) = 0.561$  to  $F(1) = 0.634$ , a 13.01% increase with respect to the previously selected properties.

In summary, the performance of the one-step baseline is  $F(1) = 0.659$ . In contrast, figure 2 highlights the 16 out of the 25 features utilized by the second phase of NERQ-2S. Note that the “new” bar indicates the token/entity attribute determined in the first step. Most importantly, NERQ-2S finished with an  $F(1) = 0.729$ , which means a 10.62% enhancement with respect to the one-step baseline. From these results, it is worth considering the following aspects:

1. In terms of features, 11 of the 13 attributes used by the one-step baseline were also exploited by NERQ-2S. Further, NERQ-2S profits from four additional properties that were also available for the one-step baseline.
2. The five more prominent properties selected by the baseline, were also chosen by NERQ-2S with just a slight change in order.
3. The “new” feature achieves an improvement of 23.51% ( $F(1) = 0.641$ ) with respect to the previous selected property. The impact of the entity/token attribute can be measure when compared with the performance accomplished by the first five features selected by the baseline ( $F(1) = 0.634$ ).

In light of these results, we can conclude that: a) adding the entity/token feature to the CRF is vital for boosting the performance, making a two-step approach a better solution than the traditional one-step approach; and b) this entity/token property is complementary to the list shown in table 2.

The confusion matrix for NERQ-2S shows that errors, basically, regard highly ambiguous terms. Some interesting misclassifications:

1. Overall, 17.38% of the terms belonging to place names were mistagged by NERQ-2S. From these, 72.11% were perceived as part of organization names.
2. On the other hand, 17.27% of the terms corresponding to organization names were mislabelled by NERQ-2S. Here, 15.52% and 12.84% of these errors were due to the fact that these terms were seen as tokens and parts of place names, respectively.

ID	Feature	Example
Word Features		
0	Current term ( $t_i$ )	abc123
1	Previous term ( $t_{i-1}$ )	before
2	Next word ( $t_{i+1}$ )	after
N-grams		
3	Bi-gram of $t_{i-1}$ and $t_i$	before abc123
4	Bi-gram of $t_i$ and $t_{i+1}$	abc123 after
Pre- & Postfix		
5	1 leftmost character from $t_i$	a
6	2 leftmost characters from $t_i$	ab
7	3 leftmost characters from $t_i$	abc
8	1 rightmost character from $t_i$	3
9	2 rightmost characters from $t_i$	23
10	3 rightmost characters from $t_i$	123
Reductions		
11	$t_i$ without digits	abc
12	$t_i$ without letters	123
Word Shape		
13	Shape of $t_i$ (“a” represents letters; “0” digits, “-” special characters)	aaa000
14	Shape of $t_i$ (same elements joined)	a0
Position & Lengths		
15	Position of $t_i$ from left	3
16	Position of $t_i$ from right	2
17	Character length of $t_i$	6
Boolean		
18	$t_i$ is a number? (only digits)	false
19	$t_i$ is a word? (only letters)	false
20	$t_i$ is a mixture of letters and digits?	true
21	$t_i$ contains “?”	false
22	$t_i$ contains apostrophe?	false
23	$t_i$ contains other special characters?	false

Table 2: List of used features. Examples are for the third term of query “*first before abc123 after*”.

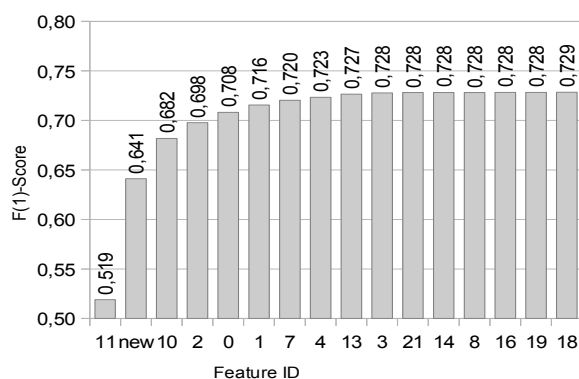


Figure 2: Attributes selected by the greedy algorithm and their respective contribution (NERQ-2S). See also table 2 for id-feature mappings. The word “new” denotes the binary token/entity attribute determined in the first step.

Incidentally, NERQ-2S mislabelled 10.40% of the tokens (non-named entity terms), while the one-step baseline 17.57%. This difference signals the importance of first-step consisting of an specialized and efficient token/entity term annotator. With regard to the first step of NERQ-2S, nine out of the 24 properties were useful, and the first step finished with an  $F(1) = 0.8077$ . From these nine attributes, eight correspond to the top eight features used by our one-step baseline, and one extra attribute (id=20). Thus, the discriminative probabilistic model learned in this first step is more specialized for this task. That is to say, though the context of a term might be modelled similarly, the parameters of the CRF model are different.

The confusion matrix for this binary classifier shows that 11.44% of entity terms were mistagged as token, while 22.24% of tokens as entity terms. This means a higher percentage of errors comes from mislabelled tokens.

On a final note, as a means of quantifying the impact of the first step on NERQ-2S, we replaced the output given by the first CRF model with the manual binary token/annotations given by the editorial team. In other words, the “new” feature is now a manual input instead of an automatically computed property. By doing this, NERQ-2S increases the performance from  $F(1) = 0.729$  to  $F(1) = 0.809$ , which means 10.97% better than NERQ-2S and 22.76% than the one-step baseline. This corroborates that a two-step approach to NERQ is promising.

## 5 Conclusions and Further Work

This paper presents NERQ-2S, a two-step approach to the problem of recognizing named entities in search queries. In the first stage, NERQ-2S checks as to whether or not each query term belongs to a named entity, and in the second phase, it categorizes each token according to a set of pre-defined classes. These classes are aimed at enhancing the user experience with the search engine in contrast to previous pre-defined categories.

Our results indicate that our two-step approach outperforms the typical one-step NERQ. Since our error analysis indicates that there is about 11% of potential global improvement by boosting the performance of the entity/token tagger, one research direction regards combining the output of distinct two-sided classifiers for improving the overall performance of NERQ-2S.

## References

- Areej Alasiry, Mark Levene, and Alexandra Poulouvasilis. 2012a. Detecting candidate named entities in search queries. In *SIGIR*, pages 1049–1050.
- Areej Alasiry, Mark Levene, and Alexandra Poulouvasilis. 2012b. Extraction and evaluation of candidate named entities in search engine queries. In *WISE*, pages 483–496.
- Junwu Du, Zhimin Zhang, Jun Yan, Yan Cui, and Zheng Chen. 2010. Using search session context for named entity recognition in query. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*.
- Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*, page 267, New York, New York, USA. ACM Press.
- Jian Hu, Gang Wang, Fred Lochovsky, Jian-Tao Sun, and Zheng Chen. 2009. Understanding users query intent with Wikipedia. In *Proceedings of WWW-09*.
- A. Jain and Marco Pennacchiotti. 2010. Open entity extraction from web search query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 510–518.
- David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January. Publisher: John Benjamins Publishing Company.
- Marius Paşca. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*, page 683, New York, New York, USA. ACM Press.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xiaoxin Yin and Sarthak Shah. 2010. Building Taxonomy of Web Search Intents for Name Entity Queries. In *Proceedings of WWW-2010*.

# A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction

Florian Boudin

LINA - UMR CNRS 6241, Université de Nantes, France

florian.boudin@univ-nantes.fr

## Abstract

In this paper, we present and compare various centrality measures for graph-based keyphrase extraction. Through experiments carried out on three standard datasets of different languages and domains, we show that simple degree centrality achieve results comparable to the widely used TextRank algorithm, and that closeness centrality obtains the best results on short documents.

## 1 Introduction

Keyphrases are the words and phrases that precisely and compactly represent the content of a document. Keyphrases are useful for a variety of tasks such as summarization (Zha, 2002), information retrieval (Jones and Staveley, 1999) and document clustering (Han et al., 2007). However, many documents do not come with manually assigned keyphrases. This is because assigning keyphrases to documents is very costly and time consuming. As a consequence, automatic keyphrase extraction has attracted considerable attention over the last few years.

Previous works fall into two categories: supervised and unsupervised methods. The idea behind supervised methods is to recast keyphrase extraction as a binary classification task (Witten et al., 1999). Unsupervised approaches proposed so far have involved a number of techniques, including language modeling (Tomokiyo and Hurst, 2003), clustering (Liu et al., 2009) and graph-based ranking (Mihalcea and Tarau, 2004). While supervised approaches have generally proven more successful, the need for training data and the bias towards the domain on which they are trained remain two critical issues.

In this work, we focus on graph-based methods for keyphrase extraction. Given a document, these

methods construct a word graph from which the most important nodes are selected as keyphrases. TextRank (Mihalcea and Tarau, 2004), a ranking algorithm based on the concept of eigenvector centrality, is usually applied to compute the importance of the nodes in the graph. Here, centrality is used to estimate the importance of a word in a document.

The concept of centrality in a graph has been extensively studied in the field of social network analysis and many different measures were proposed, see (Opsahl et al., 2010) for a review. Surprisingly, very few attempts have been made to apply such measures to keyphrase extraction. (Litvak et al., 2011) is one of them, where degree centrality is used to select keyphrases. However, they evaluate their method indirectly through a summarization task, and to our knowledge there are no published experiments using other centrality measures for keyphrase extraction. In this study, we conduct a systematic evaluation of the most well-known centrality measures applied to the task of keyphrase extraction on three standard evaluation datasets of different languages and domains<sup>1</sup>.

The rest of this paper is organized as follows. We first briefly review the previous work, followed by a description of the centrality measures. Next, we present our experiments and results and conclude with a discussion.

## 2 Related work

Graph-based keyphrase extraction has received much attention recently and many different approaches have been proposed (Mihalcea and Tarau, 2004; Wan and Xiao, 2008a; Wan and Xiao, 2008b; Liang et al., 2009; Tsatsaronis et al., 2010; Liu et al., 2010). All of these approaches use a graph representation of the documents in

<sup>1</sup>Code and datasets used in this study are available at [https://github.com/boudinfl/centrality\\_measures\\_ijcnlp13](https://github.com/boudinfl/centrality_measures_ijcnlp13)

which nodes are words or phrases, and edges represent co-occurrence or semantic relations. The importance of each node is computed using TextRank (Mihalcea and Tarau, 2004), a graph-based ranking algorithm derived from Google’s PageRank (Page et al., 1999). Words corresponding to the top ranked nodes are then selected and assembled to generate keyphrases.

Most previous studies focus on building a more accurate graph representation from the content of the documents (Tsatsaronis et al., 2010) or adding features to TextRank (Liu et al., 2010), but very few tried to use other existing centrality measures. The only works we are aware of are that of Litvak and Last (2008) that applied the HITS algorithm (Kleinberg, 1999), and Litvak et al. (2011) in which TextRank and degree centrality are compared. However, both works were evaluated against a summarization dataset by checking whether extracted keyphrases appear in reference summaries. This methodology is somewhat unreliable, as a word that occurs in a summary is not necessarily a keyphrase (e.g. experiments, results).

### 3 Keyphrase extraction

Extracting keyphrases from a document can be divided into three steps. First, a word graph is constructed from the document. The importance of each word is then determined using a centrality measure. Lastly, keyphrase candidates are generated and ranked based on the words they contain. The following sections describe each of these steps in detail.

#### 3.1 Graph construction

Given a document, the first step consists in building a graph representation from its content. An undirected word graph is constructed for each document, in which nodes are words and edges represent co-occurrence relations within a window of maximum  $N$  words. Words added to the graph are restricted with syntactic filters, which select only lexical units of a certain Part-of-Speech (nouns and adjectives). Edges are weighted according to the co-occurrence count of the words they connect. Following (Wan and Xiao, 2008b), we set the co-occurrence window size to 10 in all our experiments.

#### 3.2 Centrality measures

Once the word graph is constructed, centrality measures are computed to assign a score to each node. Let  $G = (V, E)$  be a graph with a set of vertices (nodes)  $V$  and a set of edges  $E$ . Starting with degree centrality, this section describes the ranking models we will be using in this study.

**Degree centrality** is defined as the number of edges incident upon a node. Applied to a word graph, the degree of a node  $V_i$  represents the number of words that co-occur with the word corresponding to  $V_i$ . Let  $\mathcal{N}(V_i)$  be the set of nodes connected to  $V_i$ , the degree centrality of a node  $V_i$  is given by:

$$C_D(V_i) = \frac{|\mathcal{N}(V_i)|}{|V| - 1} \quad (1)$$

**Closeness centrality** is defined as the inverse of farness, i.e. the sum of the shortest distances between a node and all the other nodes. Let  $\text{distance}(V_i, V_j)$  be the shortest distance between nodes  $V_i$  and  $V_j$  (in our case, computed using inverted edge weights to use co-occurrence information), the closeness centrality of a node  $V_i$  is given by:

$$C_C(V_i) = \frac{|V| - 1}{\sum_{V_j \in V} \text{distance}(V_i, V_j)} \quad (2)$$

**Betweenness centrality** quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. Let  $\sigma(V_j, V_k)$  be the number of shortest paths from node  $V_j$  to node  $V_k$ , and  $\sigma(V_j, V_k|V_i)$  the number of those paths that pass through node  $V_i$ . The betweenness centrality of a node  $V_i$  is given by:

$$C_B(V_i) = \frac{\sum_{V_j \neq V_i \neq V_k \in V} \frac{\sigma(V_j, V_k|V_i)}{\sigma(V_j, V_k)}}{(|V| - 1)(|V| - 2)/2} \quad (3)$$

**Eigenvector centrality** measures the centrality of a node as a function of the centralities of its neighbors. Unlike degree, it accounts for the notion that connections to high-scoring nodes are more important than those to low-scoring ones. Let  $w_{ji}$  be the weight of the edge between nodes  $V_j$  and  $V_i$  and  $\lambda$  a constant, the eigenvector centrality of a node  $V_i$  is given by:

$$C_E(V_i) = \frac{1}{\lambda} \sum_{V_j \in \mathcal{N}(V_i)} w_{ji} \times C_E(V_j) \quad (4)$$



**TextRank** is based on the eigenvector centrality measure and implements the concept of “voting”. Let  $d$  be a damping factor (set to 0.85 as in (Mihalcea and Tarau, 2004)), the TextRank score  $S(V_i)$  of a node  $V_i$  is initialized to a default value and computed iteratively until convergence using the following equation:

$$S(V_i) = (1-d) + \left( d \times \sum_{V_j \in \mathcal{N}(V_i)} \frac{w_{ji} \times S(V_j)}{\sum_{V_k \in \mathcal{N}(V_j)} w_{jk}} \right) \quad (5)$$

### 3.3 Keyphrase selection

Selecting keyphrases is a two step process. First, keyphrase candidates are extracted from the document. Sequences of adjacent words, restricted to nouns and adjectives only, are considered as candidates. Extracting sequences of adjacent words instead of n-grams ensure that keyphrase candidates are grammatically correct but entail a lower recall.

The score of a candidate keyphrase  $k$  is computed by summing the scores of the words it contains normalized by its length + 1 to favor longer n-grams (see equation 6).

$$\text{score}(k) = \frac{\sum_{\text{word} \in k} \text{Score}(\text{word})}{\text{length}(k) + 1} \quad (6)$$

Keyphrase candidates are then ranked and redundant candidates filtered out. Two candidates are considered redundant if they have a same stemmed form (e.g. “precisions” and “precision” are both stemmed to “precis”).

## 4 Experimental settings

### 4.1 Datasets

As mentioned by (Hasan and Ng, 2010), it is essential to evaluate keyphrase extraction methods on multiple datasets to fully understand their strengths and weaknesses. Accordingly, we use three different datasets in our experiments. An overview of each dataset is given in Table 1.

The **Inspecc** dataset (Hulth, 2003) is a collection of abstracts from journal papers. We use the 500 abstracts designated as the test set and the set of uncontrolled keyphrases.

The **Semeval** dataset (Kim et al., 2010) is composed of scientific articles collected from the ACM Digital Library. We use the 100 articles of the test set and its set of combined author- and reader-assigned keyphrases.

The **DEFT** dataset (Paroubek et al., 2012) is made of French scientific articles published in social science journals. We use the 93 articles of the test set and its set of author-assigned keyphrases.

	<b>Inspecc</b>	<b>Semeval</b>	<b>DEFT</b>
Type	abstracts	articles	articles
Language	English	English	French
Documents	500	100	93
Tokens/document	136	5180	6970
Keyphrases/document	9.8	14.7	5.2
Tokens/keyphrase	2.3	2.1	1.6

Table 1: Overview of the three datasets we use in our experiments.

### 4.2 Pre-processing

For each dataset, we apply the following pre-processing steps: sentence segmentation, tokenisation and Part-of-Speech tagging. For the latter, we use the Stanford POS-tagger (Toutanova et al., 2003) for English and MELt (Denis and Sagot, 2009) for French. We use the networkx<sup>2</sup> package to compute the centrality measures.

### 4.3 Evaluation measures

The performance of each centrality measure is evaluated with precision, recall and f-score at the top 10 keyphrases. Candidate and reference keyphrases are stemmed to reduce the number of mismatches. Consistent with (Hasan and Ng, 2010), we also report the performance of each centrality measure in terms of precision-recall curves for the three datasets. To generate the curves, we vary the number of extracted keyphrases from 1 to the total number of keyphrase candidates.

## 5 Results

Table 2 presents the performance of each centrality measure on the three datasets. Overall, we observe that the best results are obtained using degree which is the simplest centrality measure both conceptually and computationally. Closeness obtains the best results on Inspecc and significantly outperforms TextRank. However, it yields the worst performance on the other two datasets. This suggests that closeness is best suited for short documents (Inspecc documents are 136 tokens long on average).

<sup>2</sup><http://networkx.github.io/>

Centrality	Inspec			Semeval			DEFT		
	P	R	F	P	R	F	P	R	F
Degree	31.4	37.6	32.2	<b>11.4</b>	<b>8.0</b>	<b>9.3</b>	<b>7.7</b>	<b>14.8</b>	<b>10.0</b>
Closeness	<b>32.8<sup>‡</sup></b>	<b>38.6<sup>†</sup></b>	<b>33.3<sup>‡</sup></b>	4.1	2.8	3.3	2.6	5.2	3.4
Betweenness	31.5	37.7	32.3	10.0	7.1	8.2	7.3	13.9	9.5
Eigenvector	29.5	35.0	30.2	10.7	7.4	8.7	6.2	12.1	8.1
TextRank	31.5	37.7	32.2	10.7	7.4	8.7	7.6	14.5	9.9

Table 2: Performance of each centrality measure in terms of precision, recall and f-score at the top 10 keyphrases on the three datasets (<sup>†</sup> and <sup>‡</sup> indicate a significant improvement over TextRank at the 0.05 and 0.01 levels respectively using Student’s t-test).

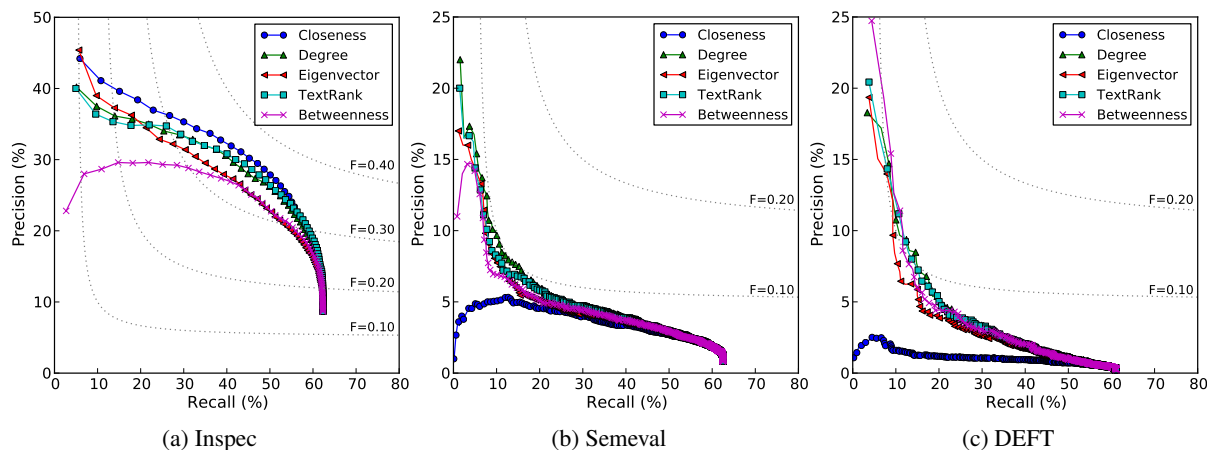


Figure 1: Precision-recall curves for each centrality measure on the three datasets.

To get a better understanding of the performance for each centrality measure, we report in Figure 1 their precision-recall curves for each of the three datasets. Moreover, to estimate how each measure performs in terms of f-score, we also plot the curves corresponding to different levels of f-score. Again, we observe that the best measure for the Inspec dataset is closeness. For the other two datasets, there is no centrality measure which overall performs best. We note that the maximum recall is almost the same for the three datasets.

Interestingly, degree and TextRank achieve similar performance on the three datasets. The reason for this is that TextRank is derived from PageRank which was shown to be proportional to the degree distribution for undirected graphs (Grolmusz, 2012). Degree centrality, whose time complexity is  $\Theta(V^2)$ , can therefore advantageously replace TextRank for keyphrase extraction.

## 6 Conclusion

In this paper, we presented a comparison of five centrality measures for graph-based keyphrase extraction. Using three standard datasets of different languages and domains, we showed that degree centrality, despite being conceptually the simplest measure, achieves results comparable to the widely used TextRank algorithm. Moreover, results show that closeness significantly outperforms the other centrality measures on short documents.

## Acknowledgments

The author would like to thank Emmanuel Morin and Solen Quiniou for their helpful comments on this work. We also thank the anonymous reviewers for their useful comments. This work was supported by the French Agence Nationale de la Recherche under grant ANR-12-CORD-0029 (TermITH project).

## References

- Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *Proceedings of PACLIC 2009*, pages 110–119.
- Vince Grolmusz. 2012. A note on the pagerank of undirected graphs. *CoRR*, abs/1205.1960.
- Juhyun Han, Taehwan Kim, and Joongmin Choi. 2007. Web document clustering by using automatic keyphrase extraction. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, pages 56–59.
- Kazi Saidul Hasan and Vincent Ng. 2010. Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In *Proceedings of COLING 2010: Posters*, pages 365–373.
- Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of EMNLP 2003*, pages 216–223.
- Steve Jones and Mark S. Staveley. 1999. Phrasier: a system for interactive document retrieval using keyphrases. In *Proceedings of SIGIR 1999*, pages 160–167.
- Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26.
- Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.
- Weiming Liang, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2009. Extracting keyphrases from chinese news articles using textrank and query log knowledge. In *Proceedings of PACLIC 2009*, pages 733–740.
- Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24.
- Marina Litvak, Mark Last, Hen Aizenman, Inbal Gubits, and Abraham Kandel. 2011. DegExt — A Language-Independent Graph-Based Keyphrase Extractor. In *Advances in Intelligent Web Mastering*, pages 121–130. Springer.
- Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of EMNLP 2009*, pages 257–266.
- Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of EMNLP 2010*, pages 366–376.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pages 404–411.
- Tore Opsahl, Filip Agneessens, and John Skvoretz. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: bringing order to the web.
- Patrick Paroubek, Pierre Zweigenbaum, Dominic Forest, and Cyril Grouin. 2012. Indexation libre et contrôlée d’articles scientifiques. Présentation et résultats du défi fouille de textes DEFT2012. In *Proceedings of the DÉfi Fouille de Textes 2012 Workshop*, pages 1–13.
- Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL 2003*, pages 173–180.
- George Tsatsaronis, Iraklis Varlamis, and Kjetil Nørvåg. 2010. Semanticrank: Ranking keywords and sentences using semantic graphs. In *Proceedings of COLING 2010*, pages 1074–1082.
- Xiaojun Wan and Jianguo Xiao. 2008a. Colabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of COLING 2008*, pages 969–976.
- Xiaojun Wan and Jianguo Xiao. 2008b. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence, AAAI’08*, pages 855–860.
- Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255.
- Hongyuan Zha. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of SIGIR 2002*, pages 113–120.

# Translating Chinese Unknown Words by Automatically Acquired Templates

Ming-Hong Bai<sup>1,2</sup>   Yu-Ming Hsieh<sup>1,2</sup>   Keh-Jiann Chen<sup>1</sup>   Jason S. Chang<sup>2</sup>

<sup>1</sup> Institute of Information Science, Academia Sinica, Taiwan

<sup>2</sup> Department of Computer Science, National Tsing-Hua University, Taiwan

mhbai@sinica.edu.tw, morris@iis.sinica.edu.tw,

kchen@iis.sinica.edu.tw, jason.jschang@gmail.com

## Abstract

In this paper, we present a translation template model to translate Chinese unknown words. The model exploits translation templates, which are extracted automatically from a word-aligned parallel corpus, to translate unknown words. The translation templates are designed in accordance with the structure of unknown words. When an unknown word is detected during translation, the model applies translation templates to the word to get a set of matched templates, and then translates the word into a set of suggested translations. Our experiment results demonstrate that the translations suggested by the unknown word translation template model significantly improve the performance of the Moses machine translation system.

## 1 Introduction

Automatic translation of unknown words is still an open problem. As a result, most statistical machine translation (SMT) systems treat such words as unknown tokens and leave them untranslated. (Koehn et al., 2003; Chiang, 2005; Koehn et al., 2007)

The unknown word translation problem has generated considerable interest in recent years. Some works (e.g., Callison-Burch et al., 2006; Marton et al., 2009; Mirkin et al., 2009) focus on finding in-vocabulary paraphrases, which are then used as bridges to translate target unknown words. Li and Yarowsky (2008) proposed an unsupervised method for extracting the mappings from Chinese abbreviations and their full-forms. The method exploits the full-forms as bridges to translate the abbreviations. A prerequisite of the above methods is that the unknown words must have paraphrases (or full-forms). However, many

types of unknown words do not have paraphrases (full-forms) naturally.

In contrast to paraphrasing methods, Huang *et al.* (2011) developed a sublexical translation method that translates an unknown word by combining the translations of its sublexicals. However, to deal with the reordering problem, the model combines the translations of sublexicals by considering both *straight* and *inverse* directions and uses a language model to select the better one. The ordering is generally morphological structure dependent, but language models only select the most fluent order without considering morphological constraints.

In this paper, we propose a translation template model to translate Chinese unknown words. Our model has a number of advantages. First, the translation templates can be extracted automatically from a word-aligned parallel corpus. Second, the word order information is encoded in the templates, so we can compose the translation of an unknown word in a more reliable order. Finally, the expansion of the non-terminal symbol in the translation templates is flexible.

The remainder of this paper is organized as follows. In the next section, we introduce the proposed translation template model. In Section 3, we describe the experimental setup; and in Section 4, we evaluate the translations of unknown words derived by our model. Section 5 contains some concluding remarks.

## 2 Translation Template Model

The form of a translation template is similar to that of the *hierarchical phrase pair* rule (Chiang, 2005), except that the translation template is designed for translating unknown words, whereas the hierarchical phrase pair rule is designed for translating phrases. As a result, they differ in terms of the training process and rule fitting process.

---

$Na \rightarrow < [Na_1]業, [Na_1] industry >$   
 $Nc \rightarrow < [Nc_1]市, [Nc_1] city >$   
 $Na \rightarrow < 副[Na_1], deputy [Na_1] >$   
 $Nc \rightarrow < [Nv_1] 司, secretary for [Nv_1] >$

---

Figure 1. Examples of translation templates; the notations in brackets represent the non-terminal symbols.

As shown in Figure 1., a translation template is comprised of three parts: a non-terminal symbol ( $Na$ ) on the left-hand side, a source language template ( $[Na_1]業$ ) in the middle, and a target language template ( $[Na_1] industry$ ) on the right-hand side.

## 2.1 Definition of Translation Template

Based on the symbols used by Chiang (2005), we define a translation template as follows:

$$X \rightarrow < \gamma, \alpha, \sim > \quad (1)$$

where  $X$  is a left-hand side non-terminal symbol, which is usually a part-of-speech that constrains the part-of-speech of the target unknown word;  $\gamma$  is a translation template of the source language, and may contain terminal and non-terminal symbols;  $\alpha$  is a translation template of the target language, and may also contain terminal and non-terminal symbols; and  $\sim$  is a one-to-one correspondence between non-terminal occurrences in  $\gamma$  and non-terminal occurrences in  $\alpha$ .

## 2.2 Translation Process

The steps of the translation process for a given unknown word are as follows:

- Apply translation templates to the unknown word and return the matched templates.
- Translate the word based on the matched templates.
- Compute the scores for each translation candidate.

We take "出口業" (export industry) as an example to illustrate the translation process. First, translation templates are applied to the word and a set of templates are returned (shown as Figure 2).

---

$Na \rightarrow < [Nv_1]業, [Nv_1] industry >$   
 $Na \rightarrow < [Nv_1]業, [Nv_1] business >$

---

Figure 2. The matched translation templates.

Then, the non-terminal symbol of each rule is expanded with the translation equivalents of the

in-vocabulary word "出口" (export) and the following translation candidates are generated by the matched translation templates (shown as Figure 3).

---

$Na \rightarrow < 出口業, export industry >$   
 $Na \rightarrow < 出口業, exportation industry >$   
 $Na \rightarrow < 出口業, export business >$   
 $Na \rightarrow < 出口業, exportation business >$

---

Figure 3. Translation candidates.

In the final step, we compute each translation candidate's score, and then rank all the candidates to drive the top-n translations.

## 2.3 Translation Probability and Lexical Weighting

Most phrase-based SMT systems use the *translation probability* and the *lexical weighting* as the parameters of scoring functions for translated phrases (Koehn et al., 2003). The original SMT translation probability is defined as follows:

$$p(\mathbf{f} | \mathbf{e}) = \frac{freq(\mathbf{f}, \mathbf{e})}{freq(\mathbf{e})} \quad (2)$$

where  $\mathbf{e}$  denotes a phrase in the source language,  $\mathbf{f}$  denotes a phrase in the target language, and  $freq(\bullet)$  denotes the frequency function.

Due to the lack of unknown words in the training data, we approximate the translation probability by using the *rule fitting probability*, which is defined as follows:

$$p(\mathbf{f} | \mathbf{e}) \cong p(X \rightarrow < \gamma, \alpha, \sim > | \mathbf{e}) \quad (3)$$

In our experiments, we utilized the maximum entropy model (Berger et al., 1996) to model the *rule fitting probability*. It is also difficult to estimate the *lexical weighting* for the translation candidates of an unknown word. The original lexical weighting is defined as follows:

$$p_w(\mathbf{f} | \mathbf{e}, \mathbf{a}) = \prod_{i=1}^n \frac{1}{|\{j | (i, j) \in \mathbf{a}\}|} \sum_{\forall (i, j) \in \mathbf{a}} p(f_i | e_j) \quad (4)$$

where  $f_i$  denotes a word in the source phrase, and  $e_j$  denotes the words in the target phrase.

For convenience, we assume that the alignment units of the unknown words are Chinese characters, and that the alignments

between Chinese characters and English words are fully linked. Under this assumption, the lexical weighting can be simplified as follows:

$$p_w(\mathbf{f} | \mathbf{e}, a) \cong \prod_{\forall f_i \in \mathbf{f}} \frac{1}{|\mathbf{e}|} \sum_{\forall c_j \in \mathbf{e}} p(f_i | c_j) \quad (5)$$

where  $c_i$  denotes a character in the source phrase  $\mathbf{e}$  (a Chinese unknown word), and  $f_i$  denotes a word in the target phrase  $\mathbf{f}$  (an English phrase).

## 2.4 Extraction of Translation Templates

The translation templates are automatically extracted from a word-aligned corpus by the following steps:

- Mark the known translation equivalents in corresponding phrase pairs in the word-aligned corpus.
- Transform the marked translation equivalent pairs into the translation template form.
- Collect the translation templates derived in the previous step and compute the frequency for each rule.

In the first step, to mine translation templates from the word-aligned corpus, we utilized multi-syllabic Chinese compound words to derive translation templates by marking their translation equivalents in the word-aligned corpus, as shown in Figure 4.

POS	Chinese $\mathbf{f}$	Aligned English $\mathbf{e}$
Na	[旅遊]業	[tourism] industry
Na	副[廠長]	deputy [director]
Nc	[運輸]司	secretary for [transport]

Figure 4. Examples of word-aligned pairs ( $\mathbf{f}$ ,  $\mathbf{e}$ ) with marked translation equivalents in the square brackets.

In the second step, we transform the marked items into the translation template form by replacing the marked words/morphemes with non-terminal symbols. The symbols on the left-hand side are part-of-speech constraints on the unknown word. Figure 5 shows the translation templates derived from the word-aligned pairs in Figure 4.

Na	→	< [Na <sub>1</sub> ]業 , [Na <sub>1</sub> ] industry >
Na	→	< 副[Na <sub>1</sub> ] , deputy [Na <sub>1</sub> ] >
Nc	→	< [Nv <sub>1</sub> ] 司 , secretary for [Nv <sub>1</sub> ] >

Figure 5. The translation templates transformed from the word-aligned pairs in Figure 4.

Finally, we collect the translation templates from the translation template tagged corpus and remove low frequency templates from the list.

## 2.5 Rule Fitting Probability

We employ the Maximum Entropy Toolkit (Zhang, 2004) to construct the rule fitting probability model, which uses the features shown in Figure 6.

POS	POS of the word.
Prefix	First character of the word.
Suffix	Last character of the word.
Character	Each character of the word.
Length	Length of the word.
Has_surname	Does the word begin with a Chinese surname?
Has_number	Does the word contain a digital number?
POS-1	POS of the previous word.
POS+1	POS of the next word.
Word-1	Previous word.
Word+1	Next word.

Figure 6. The extra features used by the *rule fitting probability* model.

## 2.6 Morphological Translation Rules

Some unknown words cannot be composed with simple morphemes. For example, "百分之八十" (80 percent) has a numeric morpheme, "八十" (80), which is not enumerable. The template model is flexible to be extended to use morphological translation rules instead of translation table to generate the translations of morphemes. We use two types of morphological translation rules: *numerical* and *phonetic* morphological translation rules.

## 3 Experimental Setting

We evaluate the model on Moses (Koehn et al., 2007) by embedding the translations of the unknown words to test data as suggestion translations.

### 3.1 Baseline SMT System and Data Sets

We used the Hong Kong Parallel Text (LDC2004T08) as the training data for the Moses SMT system and our template model. The Chinese sentences were pre-processed by the CKIP Chinese word segmentation system (Ma and Chen, 2003). The language model was trained on the English Gigaword corpus (LDC2003T05). We randomly selected 340

sentences from the NIST MT08 test data as our development set, the NIST MT06 test data and the rest of the NIST MT08 as our test set.

### 3.2 Training

The parallel text was word-aligned by the GIZA++ toolkit (Och and Ney, 2003). Then, we utilized the word-aligned corpus to extract translation templates. This process yielded a set of translation templates and a translation template tagged corpus, which was used to train the fitting probability model. To evaluate the fitting probability model, the translation template tagged corpus was randomly split into two parts to obtain a translation template tagged training set (about 1,800,000 sentences) and a translation template tagged test set (about 200,000 sentences).

We used the translation template tagged training set to train the rule fitting probability model. Then, we used the translation template tagged test set as a pseudo gold standard to evaluate the performance of the rule fitting probability model.

We also rebuilt the experiments based on the FBIS Parallel Text (LDC2003E14), which contains about 300,000 parallel sentences to verify the stability of our model. The rebuilding process is the same as that for the Hong Kong Parallel Text.

## 4 Experimental Results

We evaluated the translation template model on the NIST MT06 test set and NIST 08 subset. During the evaluation, the test sets were translated by the Moses SMT system with/without the embedded translation suggestions derived by the translation template model. The parameters in Moses were tuned by *minimum-error-rate* training (Och, 2003) on the development set.

	MT06	MT08_sub
Baseline	23.36	19.36
Trans. table	23.47 (+0.11)	19.46 (+0.10)
Phonetic	23.83 (+0.47)	19.65 (+0.29)
Numeric	23.43 (+0.07)	19.44 (+0.08)
All	23.89 (+0.53)	19.80 (+0.44)

Table 1. Evaluation results based on the Hong Kong Parallel Text.

As mentioned in Section 2.6, the translation templates can be flexible expanded by translation table as well as by morphological translation

rules. In our experiments, we exploit phonetic and numerical morphological translation rules to generate translations of morphemes. Table 1 shows the performances of the translation results with/without unknown word translation suggestions. As it shows, all of the translation expansion methods significantly improved the underlying SMT system.

To verify the stability of this method, we also rebuilt a baseline system and an unknown word translation model based on the FBIS parallel corpus, as shown in Table 2.

	MT06	MT08_sub
Baseline	24.38	19.94
Trans. table	24.54 (+0.16)	20.21 (+0.27)
Phonetic	24.78 (+0.40)	20.28 (+0.34)
Numeric	24.64 (+0.26)	20.09 (+0.15)
All	25.09 (+0.71)	20.65 (+0.71)

Table 2. Evaluation results based on the FBIS parallel corpus.

The improvement in the BLEU score is statistically significant ( $p < 0.01$ ) under the paired bootstrap re-sampling test (Koehn, 2004). The experimental results show that the proposed translation template model significantly improves the performance of the statistical machine translation system.

## 5 Conclusion

We have proposed a method that utilizes a translation template model to translate Chinese unknown words. The translation templates can be automatically extracted from a word-aligned parallel corpus and evaluated without using extra information. Experimental results show that the model can suggest accurate unknown word translations for an existing SMT system and improve the translation quality.

## References

- Berger, Adam L., Stephen A. Della Pietra, Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.
- Callison-Burch, Chris, Philipp Koehn, Miles Osborne. 2006. Improved Statistical Machine Translation Using Paraphrases. In *Proc. of HLT/NAACL 2006*. pp. 17-24
- Chiang, David. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proc. of ACL 2005*. pp. 263-270.

- Huang, Chung-chi, Ho-ching Yen and Jason S. Chang. 2011. Using Sublexical Translations to Handle the OOV Problem in Machine Translation. *ACM Transactions on Asian Language Information Processing*, 10(3): Article 16.
- Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. of HLT/NAACL'03*. pp. 127-133.
- Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP'04*. pp. 388-395.
- Koehn, Philipp, 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 *Proc. of the ACL 2007 Demo and Poster*.
- Li, Zhifei and David Yarowsky. 2008. Unsupervised translation induction for Chinese abbreviations using monolingual corpora. In *Proc. of ACL 2008*. pp. 425-433.
- Ma, Wei-Yun and Keh-Jiann Chen. 2003. Introduction to CKIP Chinese word segmentation system for the first international Chinese word segmentation bakeoff. In *Proc. of the ACL Workshop on Chinese Language Processing*. pp. 168-171.
- Marton, Yuval, Chris Callison-Burch and Philip Resnik. 2009. Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In *Proc. of ACL/AFNLP 2009*. pp. 381-390.
- Mirkin, Shachar, Lucia Specia, Nicola Cancedda, Ido Dagan, Marc Dymetman and Idan Szpektor. 2009. Source-Language Entailment Modeling for Translating Unknown Terms. In *Proc. of ACL/AFNLP 2009*. pp. 791-799.
- Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL 2003*. pp. 160-167.
- Och, Franz Josef and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, volume 29, number 1, pp. 19-51.
- Zhang, Le. 2004. Maximum entropy modeling toolkit for python and c++. available at [http://homepages.inf.ed.ac.uk/lzhang10/maxent\\_toolkit.html](http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html).



# Multilingual Lexicon Bootstrapping - Improving a Lexicon Induction System Using a Parallel Corpus

Patrick Ziering<sup>1</sup> Lonneke van der Plas<sup>1</sup> Hinrich Schütze<sup>2</sup>

<sup>1</sup>Institute for NLP, University of Stuttgart, Germany

<sup>2</sup>CIS, University of Munich, Germany

{Patrick.Ziering, Lonneke.vanderPlas}@ims.uni-stuttgart.de

## Abstract

We address the task of improving the quality of lexicon bootstrapping, i.e., of expanding a semantic lexicon on a given corpus. A main problem of iterative bootstrapping techniques is the fact that lexicon quality degrades gradually as more and more false terms are added. We propose to exploit linguistic variation between languages to reduce this problem of semantic drift with a knowledge-lean and language-independent ensemble method. Our results on English and German show that lexicon bootstrapping benefits significantly from the multilingual symbiosis.

## 1 Introduction

High-quality semantic lexicons are an essential resource for many natural language processing (NLP) tasks like information extraction or anaphora resolution. Methods for automatically bootstrapping semantic lexicons given a seed list often struggle with lexicon accuracy decrease in higher iterations depending on corpus size (Igo and Riloff, 2009). One reason for this is *semantic drift*, which occurs when erroneous terms and/or contexts are introduced into and then dominate the iterative process (Curran et al., 2007). For instance, the ambiguity found in female names such as *Iris* and *Rose* may cause the induced terms to drift into flower names (McIntosh and Curran, 2009). Examples from the patent domain, that we are focusing on in this work, are PROCESSES that may drift into the semantic class of OBJECTS when terms such as *energy storage* and *spring coupling* are induced.

Previous work has used the cross-lingual correspondence between variations in linguistic structure and variations in ambiguity as a form of naturally occurring supervision in unsupervised learning for a number of tasks (Dagan et al., 1991;

Snyder and Barzilay, 2010). On the lexical level, cross-lingual variations proved to remedy problems related to polysemy for synonym acquisition (Van der Plas and Tiedemann, 2010) and word sense disambiguation (Lefever and Hoste, 2010).

We hypothesize that cross-lingual divergences will be preeminently suitable to remedy problems related to semantic drift in iterative bootstrapping, where lexical and structural ambiguity give rise to erroneous terms and/or contexts. Languages are not isomorphic: ambiguous terms and contexts are frequently language-specific. In our example above, the English term *energy storage* is ambiguous, however, in German, each reading has its own translation. *Energy storage* is translated with *Energiespeicher* in the OBJECT reading and *Energiespeicherung* in the PROCESS reading.

Our multilingual ensemble lexicon bootstrapping system is inspired by Basilisk (Thelen and Riloff, 2002). Previous work has addressed semantic drift in Basilisk by conflict resolution between several classes (Thelen and Riloff, 2002), by using web queries (Igo and Riloff, 2009) and by combining Basilisk in an ensemble with an SVM tagger and a coreference resolution system (Qadir and Riloff, 2012). These approaches are monolingual. Instead we use a multilingual ensemble method where the induced lexicons of several languages constrain each other.

Apart from addressing semantic drift, the multilingual setting we propose has several other advantages. First, one language may leave implicit what another expresses directly in linguistic forms. In German, common nouns are capitalized and compound nouns are written as one word. We propagate German noun information via word alignment to English and thereby learn both single words as well as most multiword expressions (MWEs) without the need for a noun chunker or MWE recognizer.

Second, as a result of the multilingual ensem-

ble method we are able to induce lexicons for any language given a parallel corpus. We do not need seed lists for all languages, which are often sparse. Translating<sup>1</sup> the English seed list automatically results in high-quality lexicons for all other languages.

Finally, many pattern-based lexicon bootstrapping methods use pre-defined patterns which require language- and domain-specific syntactic analyses. Our multilingual approach makes use of a parallel corpus and tools from phrase-based machine translation that substitute for the necessity of pattern definition and once more guarantees a knowledge-lean and language-independent process.

## 2 Multilingual Lexicon Bootstrapping

### Monolingual bootstrapping

```

1: lexicon ← seed
2: for int  $i = 0; i < m; i++$  do
3:   patterns ← patternsOf(lexicon)
4:   score(patterns)
5:   patterns ← return-top-k(patterns, 20 +  $i$ )
6:   terms ← termsOf(patterns) – lexicon
7:   score(terms)
8:   lexicon ← lexicon  $\cup$  return-top-k(terms,  $t$ )
9: end for
10: return lexicon

```

Figure 1: Basilisk (Thelen and Riloff, 2002)

Our basic algorithm is inspired by Basilisk (Thelen and Riloff, 2002), an algorithm developed for monolingual lexicon bootstrapping as shown in Figure 1. The starting point is a lexicon initialized with a given seed set. Then the lexicon is expanded iteratively. First Basilisk ranks all patterns containing terms from the lexicon (lexicon terms) (lines 3-4) based on the RlogF score:

$$\text{RlogF}(\text{pattern}_i) = F_i/N_i \log_2(F_i)$$

where  $F_i$  is the number of lexicon terms occurring in  $\text{pattern}_i$  and  $N_i$  is the total number of terms occurring in  $\text{pattern}_i$ . Then Basilisk ranks all non-lexicon terms that occur in the  $20+i$  highest-ranked patterns (where  $i+1$  is the number of performed iterations) (lines 5-7) based on the AvgLog score:

$$\text{AvgLog}(\text{term}_i) = 1/P_i \sum_{j=1}^{P_i} \log_2(F_j + 1)$$

<sup>1</sup>Potential ambiguity in translated seeds will be taken care of, because our ensemble learning prevents false terms resulting from erroneous seeds to be added to the lexicon.

where  $P_i$  is the number of patterns containing term $_i$ . Finally Basilisk adds the  $t$  (originally 5) highest-ranked terms to the lexicon (line 8) and process repeats.

### Multilingual ensemble framework

```

1: for  $L_i$  in  $\{L_1, \dots, L_n\}$  do
2:    $B_i \leftarrow$  initialize Basilisk for  $L_i$ 
3:    $B_i.\text{final} \leftarrow \{\}$ 
4: end for
5: while  $\exists i$  ( $\text{size}(B_i.\text{final}) < l$ ) do
6:   for  $B_i$  in  $\{B_1, \dots, B_n\}$  do
7:      $B_i.\text{iterate}(m, t)$ 
8:   end for
9:   consensusCheck( $\{B_1, \dots, B_n\}$ )
10: end while
11: return ( $B_1.\text{final}, \dots, B_n.\text{final}$ )

```

Figure 2: Multilingual bootstrapping

We adapted Basilisk to a multilingual setting as shown in Figure 2. Key to the framework is the multilingual consensus check. In the **consensusCheck**, for each Basilisk process  $B_i$  we intersect its lexicon with the translations of the lexicons of all other Basilisk processes  $B_k$ . We translated the lexicons from  $L_k$  to  $L_i$  using the bilingual dictionary  $\text{DICT}_{k \leftrightarrow i}$  extracted from the corresponding phrase table. If the lexicon intersection is non-empty, the consensus terms are added to the final list of  $B_i$  and the temporary lexicon is reset to the seed terms and the final list. If the intersection is empty, the lexicons are maintained completely leading to a higher chance of non-emptiness in the subsequent multilingual iteration.

We first initialize a Basilisk process  $B_i$  for each language  $L_i$  (Figure 1, line 1; Figure 2, lines 1-4). For each Basilisk process, we introduce a *final* list, that contains only lexicon terms that survived the consensus check. As long as at least one Basilisk process has a final list containing less than  $l$  terms, each Basilisk process performs  $m$  iterations of learning the top  $t$  terms<sup>2</sup> each (Figure 1, line 2-9; Figure 2, lines 6-8). Multilingual bootstrapping is finished, when all Basilisk processes have a final list of at least  $l$  terms (Figure 2, line 11).

<sup>2</sup>Thelen and Riloff (2002) originally set  $t = 5$  - this would be inefficient with our ensemble lexicon bootstrapping on state-of-the-art machines because most of the time, there would be no consensus terms. We set  $m = 2$  and  $t = 25$ , which seems to be a good trade-off between time and accuracy.

### 3 Experiments

Although our method is multilingual and language-independent we restrict our demonstration of its potential to two languages: German and English.

**The parallel corpus.** We use patent data distributed by the European Patent Office (EPO<sup>3</sup>) between 1998 and 2008. Most European patents provide their claims (the part of a patent defining the scope of protection) in German, English and French. We constructed a German-English parallel corpus out of 177,317 patent documents.

**Creation of Moses phrase table.** For each unordered language pair, we create a MOSES (Koehn et al., 2007) phrase table in several steps. We first apply sentence alignment (GARGANTUA Braune and Fraser (2010)), then word alignment (MGIZA++ Gao and Vogel (2008)) to the data. And finally, we apply the statistical machine translation tool MOSES to the parallel word-aligned data. The resulting data structure is a phrase table of word-aligned phrases in two languages as shown in Figure 3, where the third line indicates the word alignment.

Verfahren zur selektiven Flüssigphasenhydrierung
the process for selective liquid phase hydrogenation
0-0 0-1 1-2 2-3 3-4 3-5 3-6

Figure 3: Main content in a phrase table entry

**Extracting terms and patterns.** For term extraction, we define one language as the term-specifying language  $L_{term}$  (i.e., the language that specifies the set of candidate terms for all languages) – in our case, we choose German since it expresses term boundaries very directly in its linguistic forms (capitalized nouns, single word compounding). German terms are defined as a capitalized token with at least 4 letters. For each unordered language pair  $\{L_{term}, L_i\}$ , we define each term in  $L_i$  as a sequence of tokens that are aligned to a term in  $L_{term}$ . In Figure 3, “liquid phase hydrogenation” is defined as term since it is aligned to “Flüssigphasenhydrierung”.

For reducing errors due to poor word alignment<sup>4</sup>, we apply MATE (Bohnet, 2010) part of speech (PoS) tagger on phrases in languages other

<sup>3</sup>www.epo.org

<sup>4</sup>Since our corpus is not large enough for perfect word alignment, it can be supported by a part of speech tagger. To keep the process completely language-independent, this step may also be skipped.

than  $L_{term}$  and define a PoS filter that removes spurious tokens at the left and right boundaries. Figure 4 shows the PoS filter for English, that is adapted from Justeson and Katz (1995) to the task of filtering tokens. The aligned terms that pass the filtering are stored in a dictionary  $DICT_{term \leftrightarrow i}$ .

English (JJ|VBG|NN)\* NN (IN NN+)?

Figure 4: PoS pattern for term filtering

Patterns are extracted from the phrase tables as well. For each phrase in  $L_{term}$  and  $L_i$  we use the remaining tokens surrounding each term as bootstrapping pattern associated with this term (e.g., “Verfahren zur selektiven <X>” is defined as pattern for “Flüssigphasenhydrierung”).<sup>5</sup>

Our final data set contains roughly 19 million German and English term-pattern pairs. The dictionary  $DICT_{DE \leftrightarrow EN}$  comprises 1.8 million entries.

**Translating seed sets.** We define one corpus language as the seed-defining language  $L_{seed}$  – in our case, we choose English since it provides the richest lexical resources. Then, for all other languages  $L_i$  we translate each seed term from  $L_{seed}$  to  $L_i$  using the most frequent translation in  $DICT_{seed \leftrightarrow i}$ .

**Evaluation.** We evaluated the multilingual bootstrapping system on two semantic classes motivated by the technical field of patents: PROCESS and TECHNICAL QUALITY.

**PROCESS:** A method or event that results in a change of state (e.g., *stretching, molding process, redundancy control, ...*).

**TECHNICAL QUALITY:** A basic or essential attribute which is measurable or shared by all members of a group (e.g., *power consumption, piston speed, light reflection index, ...*).

The sources for the English seed sets have been WordNet lexicographer classes (Ciaramita and Johnson, 2003) and Wikipedia<sup>6</sup> word lists.

For each semantic class and language we induced lexicons of 2000 terms. For each lexicon we evaluated a sample of 200 terms. Two annotators first rated 50 terms for each language and class as TRUE or FALSE. Then they discussed disagreements. Afterwards, they rated the remaining terms in each lexicon sample. We achieved a total inter-annotator agreement of  $\kappa = .701$  (Cohen’s

<sup>5</sup>We remove unique patterns because they do not contribute to lexicon bootstrapping.

<sup>6</sup>www.wikipedia.org

Kappa). For the results, we used the labeled lexicons of the annotator that finalized the task first.

## 4 Results

In our experiments we compare two methods. The first is the monolingual bootstrapping method<sup>7</sup> and the second is the bilingual ensemble bootstrapping method. For a proper comparison both methods make use of the same data as described in Section 3. Table 1 shows the accuracy of the induced lexicons for German and English when learned separately (lines 1-2) and when induced with the bilingual ensemble bootstrapping method (line 3)<sup>8</sup>.

	Mode	Process	Technical Quality
1	DE	.730	.880
2	EN	.740	.895
3	DE / EN	.980† / .790	.960† / .955†

Table 1: Results of lexicon evaluation

Bilingual ensemble bootstrapping outperforms monolingual bootstrapping in both classes and languages. For the class TECHNICAL QUALITY there is a significant improvement in both languages (German: +.080; English: +.060). For the class PROCESS there is a significant improvement for German (+.250), whereas there is a nonsignificant improvement for English.

**Analysis and discussion.** To give the reader a better idea of how the bilingual ensemble method remedies semantic drift, we will comment on the asymmetric impact on performance, when high levels of ambiguity are present in one of the two languages.

We know from linguistic research (Ehrich and Rapp, 2000) that the German *ung*-ending is subject to sortal ambiguity. Words ending in *-ung* can be of various semantic types: processes, objects, events, and states. Many terms in the PROCESS class are described by nouns ending in *-ung*. Their sortal ambiguity gives rise to semantic drift from PROCESS to TECHNICAL QUALITY (e.g., *Belastung* can mean *charging* or *burden*), and to PROCESS-RELATED DEVICE (e.g., *Steuerung* can mean *steering* or *controller*). This sortal ambiguity of nouns in the PROCESS class does not have its

<sup>7</sup>Although the first method relies on a parallel corpus and multilingual preprocessing, we refer to it as the monolingual method because the learning is done monolingually.

<sup>8</sup>We mark each number with † if it significantly outperforms monolingual bootstrapping (z-test for proportions;  $p < .05$ ).

counterpart in the English lexicon. It is therefore not surprising that we see a large improvement in the quality of the German lexicons, when English is used in the ensemble bootstrapping method. We achieve an improvement in German of +.250, the largest improvement overall.

In the present bilingual setting, we cannot prevent the ambiguity found in the German terms to influence the English terms. We believe that this is the reason for the asymmetric impact of bilingual bootstrapping on the class PROCESS, where we see only a small improvement in English (+.060). The positive effects from ensemble learning for the English PROCESS class is partly wiped out by the influence of high levels of ambiguity in German. In future work, we plan to add several languages to be able to prevent ambiguity in one language to overshadow the multilingual ensemble.

## 5 Conclusion

We address the problem of semantic drift in iterative bootstrapping. We propose a multilingual ensemble learning method for lexicon bootstrapping, in which lexicons for several languages are induced in parallel and constrain each other. This method exploits linguistic variation between languages to reduce the impact of lexical and structural ambiguity within one language. In a case study on German and English and the two semantic classes TECHNICAL QUALITY and PROCESS, we show that bilingual lexicon bootstrapping outperforms monolingual bootstrapping in all classes and languages.

In addition, our multilingual approach to lexicon bootstrapping is particularly knowledge-lean and language-independent. A parallel corpus, language-independent machine translation tools and seed lists of one single corpus language suffice to extract patterns, determine term boundaries and provide seed lists for an in principle unlimited number of languages.

## Acknowledgments

This research was supported by the project TOPAS<sup>9</sup> (Tool platform for intelligent Patent Analysis and Summarization), its team members and the European Commission with its FP7-SME Program. The TOPAS Consortium is composed of five partners: Brüggemann Software Inc., IALE Inc., Intelisemantic Inc., Pompeu Fabra University and University of Stuttgart. We thank all colleagues for their support. This research was further supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) as part of the SFB 732. Thanks also to Sebastian Ebert for conducting the creation of the phrase tables.

<sup>9</sup>topasproject.eu

## References

- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING 2010*.
- Fabienne Braune and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *COLING '10*.
- Massimiliano Ciaramita and Mark Johnson. 2003. Super-sense tagging of unknown nouns in wordnet. In *EMNLP 2003*.
- J. R. Curran, T. Murphy, and B. Scholz. 2007. Minimising Semantic Drift with Mutual Exclusion Bootstrapping. In *PACLING 2007*.
- Ido Dagan, Alon Itai, and Ulrike Schwall. 1991. Two languages are more informative than one. In *ACL 1991*.
- Veronika Ehrich and Irene Rapp. 2000. Sortale Bedeutung und Argumentstruktur: *ung*-Nominalisierungen im Deutschen. *Zeitschrift für Sprachwissenschaft*, 19(2):245–303.
- Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*.
- Sean P. Igo and Ellen Riloff. 2009. Corpus-based semantic lexicon induction with web-based corroboration. In *NAACL 2009*, pages 18–26.
- J. Justeson and S. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, pages 9–27.
- 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 *ACL '07*.
- Els Lefever and Véronique Hoste. 2010. Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *SemEval 2010*.
- Tara McIntosh and James R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *ACL-IJCNLP 2009*.
- Ashequl Qadir and Ellen Riloff. 2012. Ensemble-based semantic lexicon induction for semantic tagging. In *\*SEM-2012*.
- Benjamin Snyder and Regina Barzilay. 2010. Climbing the tower of babel: Unsupervised multilingual learning. In *ICML 2010*.
- Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *EMNLP 2002*.
- Lonneke Van der Plas and Jörg Tiedemann. 2010. Finding medical term variations using parallel corpora and distributional similarity. In *OntoLex 2010*, Beijing, China.

# Mining Japanese Compound Words and Their Pronunciations from Web Pages and Tweets

Xianchao Wu

Baidu Inc.

wuxianchao@gmail, baidu}.com

## Abstract

Mining compound words and their pronunciations is essential for Japanese input method editors (IMEs). We propose to use a chunk-based dependency parser to mine new words, collocations and predicate-argument phrases from large-scale Japanese Web pages and tweets. The pronunciations of the compound words are automatically rewritten by a statistical machine translation (SMT) model. Experiments on applying the mined lexicon to a state-of-the-art Japanese IME system<sup>1</sup> show that the precision of Kana-Kanji conversion is significantly improved.

## 1 Introduction

New compound words are appearing everyday. Person names, technical terms and organization names are newly created and used in Web pages such as news, blogs, question-answering systems. Abbreviations, food names and event names are formed and shared in Twitter and Facebook. Mining of these new compound words, together with their pronunciations, is an important step for numerous natural language processing (NLP) applications. Taking Japanese as an example, the lexicons containing compound words (in a mixture of Kanjis and Kanas) and their pronunciations (in a sequence of Kanas) significantly influence the accuracies of speech generation (Schroeter et al., 2002) and IME systems (Kudo et al., 2011). In addition, monolingual compound words are shown to be helpful for bilingual SMTs (Liu et al., 2010).

In this paper, we mine three types (Figure 1) of new (i.e., not included in given lexicons) Japanese compound words and their pronunciations: (1) *words*, which are combinations of sin-

<sup>1</sup>freely downloadable from [www.simeji.me](http://www.simeji.me) for Android and <http://ime.baidu.jp/type/> for Windows

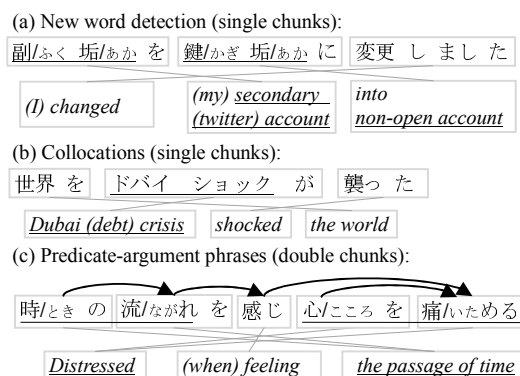


Figure 1: Examples of new (compound) words.

gle characters and/or shorter words; (2) *collocations*, which are combinations of words; and (3) *predicate-argument phrases*, which are combinations of chunks constrained by semantic dependency relations. The sentences were parsed by a state-of-the-art chunk-based Japanese dependency parser, Cabocha<sup>2</sup> (Kudo and Matsumoto, 2002a) which makes use of Mecab<sup>3</sup> with IPA dictionary<sup>4</sup> for word segmenting, POS tagging, and pronunciation annotating.

The first sentence in Figure 1 contains two new words which were not correctly recognized by Mecab. We call them “new words”, since *new* semantic meanings are generated by the combination of single characters. There is one Kana collocation in the second sentence. Different from many former researches (Manning and Schütze, 1999; Liu et al., 2009) which only mine collocations of two words, we do not limit the number of words in our collocation lexicon. The third sentence contains two predicate-argument phrases of noun-noun modifiers and object-verb relations.

The main contribution of this paper is that the

<sup>2</sup><http://code.google.com/p/cabocha/>

<sup>3</sup><http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>

<sup>4</sup><http://code.google.com/p/mecab/downloads/detail?name=mecab-ipadic-2.7.0-20070801.tar.gz>

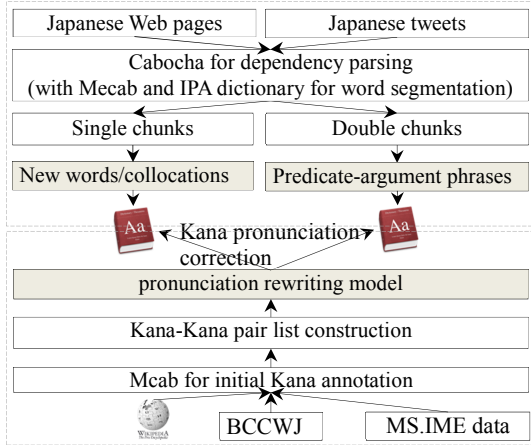


Figure 2: The lexicon mining processes.

well studied *chunk-level dependency technique* is firstly (as far as our knowledge) adapted to compound word mining. The proposed mining method has the following three parts. First, it explicitly utilize the chunk identification features and frequency information for detecting new words and collocations. Second, chunk-level semantic dependency relations are employed for determining predicate-argument phrases. Third, a Kana-to-Kana pronunciation rewriting model based on phrasal SMT framework is proposed for correcting Kana pronunciations of the compound words.

## 2 Compound Word Mining

Figure 2 shows our major lexicon mining process: lexicon mining in a top-down flow and pronunciation rewriting in a bottom-up flow.

### 2.1 Mining single chunks

**Definition 1 (Japanese chunk)** Suppose  $\mathbf{w}$  being the Japanese vocabulary set, a Japanese chunk is defined as a sequence of contiguous words,  $C = w_n^+ w_p^*$ , where  $w_n^+ \in \mathbf{w}$  is a sequence of notional words with no less than one  $w_n$ , and  $w_p^* \in \mathbf{w}$  contains zero or more particles  $w_p$ . New words and collocations come from  $w_n^+$  without  $w_p^*$ .

This mining idea is based on the fact that an Japanese morphological analyser (e.g., Mecab) tends to split one out-of-vocabulary (OOV) word into a sequence of known Kanji characters. The point is that, most of the known Kanji characters are annotated to be notional words such as nouns. Consequently, Cabocha, which takes words/characters and their POS tags as features for discriminative training using a SVM model (Kudo

	Frequency $\geq 20$	Frequency $\geq 500$
single chunk (web)	9,823,176	685,363
double chunks (web)	20,698,683	794,605
single chunk (twitter)	156,506	6,131
not in web	21,370 (13.7%)	492 (8.0%)
double chunks (twitter)	160,968	2,446
not in web	35,474 (22.0%)	443 (18.1%)

Table 1: The number of compound words mined.

and Matsumoto, 2002b), can still *correctly* tend to include these single-Kanji-character words into one chunk. Thus, we can re-combine the wrongly separated pieces into one (compound) word.

### 2.2 Mining predicate-argument phrases

**Definition 2 (Predicate-argument phrase)** A predicate-argument phrase is defined as a labelled graph structure,  $A = \langle w_h, w_n, \tau, \rho \rangle$ , where  $w_h, w_n \in \mathbf{w}$  are a predicate and an argument word (or chunk) of the dependency,  $\tau$  is a predicate type (e.g., transitive verb), and  $\rho$  is a label of the dependency of  $w_h$  and  $w_n$ . We append one constraint during mining:  $w_h$  and  $w_n$  are adjacent. That is, the phrases mined are all contiguous without gaps. The predicate-argument phrases mined in this way is helpful for context-based Kana-Kanji conversion of Japanese IME.

Japanese is a typical Subject-Object-Verb language. The direct object phrase normally appears before the verb. For example, for two input Kana sequences “やさいをいためる” (野菜/vegetables を/particle 炒める/cooking: stir-fried vegetables) and “こころをいためる” (心/heart を痛める/hurt: hurt ones heart), even “いためる” takes the similar keyboard typing, the first candidate Kanji words are totally different. The users will be angry to see the candidate of “心を炒める” (stir-fried heart) for “こころをいためる”. It is the pre-verb objects that determines the dynamic choosing of the correct Kanji verbs.

### 2.3 Experiments on compound word mining

We use two data sets for compound word mining. The first set contains 200G Japanese Web pages (1.9 billion sentences) which were downloaded by an in-house Web crawler. The second set contains 44.7 million Japanese tweets (28.8 words/tweet) which were downloaded by using an open source Java library twitter4j<sup>5</sup> which implemented the Twitter Streaming API<sup>6</sup>.

<sup>5</sup><http://twitter4j.org/ja/index.html>

<sup>6</sup><https://dev.twitter.com/docs/streaming-apis>

Lexicons	Frequency $\geq 20$	Precision
alignment method	2,562	76.5%
single chunk	16,673	93.0%
double chunks	9,099	91.5%

Table 2: The number of entries and precisions of the alignment method (Liu et al., 2009) and our approach, using 2M sentences.

Table 1 shows the statistics of the single/double chunk lexicons (of frequencies  $\geq 20$  or 500). We compared the novel entries included in the twitter lexicons but not the web. The ratio ranges from 8.0% to 22.0%, reflecting a special bag of compound words used in tweets instead of the traditional web pages.

We compare our lexicons with two baselines, one is the C-value approach (Frantzi and Ananiadou, 1999) with given POS sequences and the other is the monolingual word alignment approach (Liu et al., 2009). We ask Japanese linguists to give a POS sequence set with 128 rules for compound word mining. Applying C-value approach with these rules to the 200G web data yields a lexicon of 884,766 entries (frequency  $\geq 500$ ). Our single (double) chunk lexicon shares around 30% (7%) with this lexicon. This lexicon is used in our baseline Japanese IME system (Table 5).

During our re-implementation of the alignment approach, we found that the EM algorithm (Dempster et al., 1977) for word aligning the 1.9 billion sentences is too time-consuming. Instead, we only used the first 2M sentences (28.4 words/sentence) of the web data for intuitive comparison. The statistics are shown in Table 2. The precisions are computed by manually evaluating the top-200 entries (with higher frequencies) in each lexicon. The lexicons mined by our approach outperforms the baseline in a big distance, both precision and the number of entries successfully mined.

### 3 Pronunciation Rewriting Model

Our pronunciation rewriting model mapping from the compound words’ original pronunciations to their correct pronunciations. It is a generative model based on the phrasal SMT framework. We limit the model monotonically rewrite initial Kana sequences to their correct forms without reordering. We use Moses<sup>7</sup> (Koehn et al., 2007) to implement this model by setting the source and target sides to be Kana sequences.

<sup>7</sup><http://www.statmt.org/moses/>

The Kana-Kana rewriting model improves the traditional Kanji-Kana predication models (Hatori and Suzuki, 2011) in the following aspects. First, data sparseness problem of Kanji-Kana approach can be mitigated in a sense, since the number of Kanas in Japanese is no more than 50 yet the number of Kanjis is tens of thousands. Second, Kana-Kana pairs are easier to be aligned with each other, since most Kanjis are pronounced by no less than two Kanas and consequently the number of Kanas almost doubles the number of Kanjis in the experiment sets. Finally, the entries in the final lexicons contain two Kana pronunciations, before and after correcting. We argue this is helpful to improve the user experiences of IME systems where we need to cover the users’ typing mistakes.

#### 3.1 Mining Kanji-Kana entries from Wiki

For training the rewriting model, we mine a Kana-Kanji lexicon from parenthetical expressions in Japanese Wikipedia pages<sup>8</sup>, a high quality collection of new words. The only problem is to determine the pre-brackets Kanji sequence that exactly corresponds to the in-bracket Kana sequence.

Our method is inspired by (Okazaki and Ananiadou, 2006; Wu et al., 2009). They used a term recognition approach to build monolingual abbreviation dictionaries from English articles (Okazaki and Ananiadou, 2006) and to build Chinese-English abbreviation dictionaries from Chinese Web pages (Wu et al., 2009). For locating a textual fragment with a Kanji sequence and its Kana pronunciation in a pattern of “Kanji sequence (Kana sequence)”, we use the heuristic formula:

$$LH(c) = \text{freq}(c) - \sum_{t \in T_c} \text{freq}(t) \times \frac{\text{freq}(t)}{\sum_{t \in T_c} \text{freq}(t)}.$$

Here,  $c$  is a Kanji candidate (sub-)sequence;  $\text{freq}(c)$  denotes the frequency of co-occurrence of  $c$  with the in-brackets Kana sequence; and  $T_c$  is a set of nested Kanji sequence candidates, each of which consists of a preceding Kanji or Kana character followed by the candidate  $c$ .

Table 3 shows the number of entries mined by setting the LH score to be  $\geq 3, 4, \text{ or } 5$ . From the table, we observe that as LH threshold is added by one, the number of entries is cut nearly a half. For each entry set, we further randomly selected 200 entries and checked their correctnesses by

<sup>8</sup>All the Japanese pages until 2012.06.03 were used. Examples can be found in <http://ja.wikipedia.org/wiki/三日月>



LH $\geq$	# of Entries	Precision
3	42,423	95.0%
4	18,348	95.5%
5	10,234	96.0%

Table 3: Kanji-Kana entries mined from Wiki.

System	Prec.	BLEU-4	src/trg	Data	Train/Dev/Test
baseline	70.2%	0.8663	4.9/7.0	bcc-	25.3k/0.5k/0.5k
Ours	90.4%	0.9687	7.0/7.0	wj	
baseline	49.8%	0.6734	2.8/4.9	wiki	17.3k/0.5k/0.5k
Ours	62.2%	0.7380	4.9/4.9		
baseline	43.5%	0.9504	58.0/78.1	ms	5.6k/0.2k/0.2k
Ours	62.0%	0.9737	80.7/78.1		

Table 4: Pronunciation predication accuracies.

hand. The precisions ranges from 95% to 96%. Moreover, this mining approach can make use of parenthetical expressions appearing in not only Wikipedia but also the total Japanese Web pages.

### 3.2 Experiments on pronunciation rewriting

As shown in Figure 2, we use three data sets for training our pronunciation rewriting model. The first set is a Kanji-Kana compound lexicon collected from the 2009 Core Data of the Balanced Corpus of Contemporary Written Japanese (BC-CWJ) corpus (Maekawa, 2008). The second is the Microsoft Research IME data<sup>9</sup> (Suzuki and Gao, 2005). The third set is the Wikipedia Kana-Kanji lexicon with LH  $\geq$  4 (Table 3).

The precisions and BLEU-4 scores (Papineni et al., 2002) of the baseline system (Hatori and Suzuki, 2011) and our approach are shown in Table 4. The baseline system takes character-level translation units. From Table 4, we observe that the number of Kanas is larger than the number of Kanjis while the number of initial Kanas and corrected Kanas are almost the same. Our approach yield significant improvements ( $p < 0.01$ ) in both precisions and BLEU-4 scores.

## 4 Japanese IME Evaluation

As an application-oriented evaluation, we finally integrate the mined lexicons (as a cloud service) into a state-of-the-art Japanese IME system. The system is constructed based on the n-pos model (Mori et al., 1999; Komachi et al., 2008; Kudo et al., 2011). For training the n-pos model, we used 2.5TB Japanese Web pages as the training data. We run Mecab on Hadoop<sup>10</sup>, an open source soft-

<sup>9</sup><http://research.microsoft.com/en-us/downloads/AF99E662-B77B-4622-ADAA-7AB9F2842B20/default.aspx>

<sup>10</sup><http://hadoop.apache.org/>

IME	Top-1	Top-3	Top-5	Top-9	Test Set
baseline	38.93%	63.76%	70.47%	74.50%	twitter.net
+twitter	48.99%	70.47%	73.15%	75.17%	
baseline	50.16%	75.10%	82.99%	87.46%	JDMWE
+web	52.01%	78.61%	85.34%	89.07%	
baseline	56.23%	84.29%	91.18%	93.80%	Nagoya
+web	58.16%	84.65%	92.01%	94.35%	

Table 5: The top-n precision improvements of appending the mined twitter/web lexicons to a baseline IME system.

ware that implemented the Map-Reduce framework (Dean and Ghemawat, 2004), for parallel word segmenting and POS tagging the data.

For testifying the lexicons mined from the 200G Web data and from the tweets, we respectively use three test sets: (1) “twitter.net” with 149 entries which is a manually collected Twitter new word lexicon<sup>11</sup>; (2) partial “JDMWE” (Shudo et al., 2011) lexicon with 2,169 entries; and (3) “Nagoya” compound word lexicon<sup>12</sup> with 3,628 entries such as idioms.

The top-n (=1, 3, 5, 9) precisions are listed in Table 5. In the baseline system, we used the compound lexicon that was mined by the C-value approach using 128 POS sequences. For direct comparison, we replace this compound lexicon respectively by the web and twitter lexicons (frequency  $\geq$  500). In the twitter.net test set, the precision of the top-1 candidate significantly ( $p < 0.01$ ) improves from 38.93% to 48.99% (+10.06%). In the JDMWE and Nagoya test sets, the web lexicon can also significantly improve the top-1 precisions of around 2% ( $p < 0.05$ ). Through these numbers, we can say that the proposed approach is helpful for improving the accuracies of real-world Japanese IME application.

## 5 Conclusion

We have proposed an approach for mining new Japanese compound words from single/double chunks generated by a chunk-based dependency parser. Experiments show that the approach works well on mining new words, collocations and predicate-argument phrases from large-scale Web pages and tweets. We achieved significant improvements on top-n precisions when integrating the mined compound words together with their Kana pronunciations into a state-of-the-art Japanese IME system with million level users.

<sup>11</sup>can be downloaded from <http://netyougo.com/>

<sup>12</sup><http://kotoba.nuee.nagoya-u.ac.jp/jc2/base/list>

## Acknowledgments

The author thanks the anonymous reviewers for improving the earlier version of this paper.

## References

- Jeffrey Dean and Sanjay Ghemawat. 2004. Mapreduce: simplified data processing on large clusters. In *Proceedings of OSDI*.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38.
- Katerina T. Frantzi and Sophia Ananiadou. 1999. The c-value/nc-value domain independent method for multi-word term extraction. *Journal of Natural Language Processing*, 6:145–179.
- Jun Hatori and Hisami Suzuki. 2011. Japanese pronunciation prediction as phrasal statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 120–128, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.
- 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 ACL 2007 Demo and Poster Sessions*, pages 177–180.
- Mamoru Komachi, Shinsuke Mori, and Hiroyuki Tokunaga. 2008. Japanese, the ambiguous, and input methods (in japanese). In *Proceedings of the Summer Programming Symposium of Information Processing Society of Japan*.
- Taku Kudo and Yuji Matsumoto. 2002a. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69.
- Taku Kudo and Yuji Matsumoto. 2002b. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL-2002*, pages 63–69. Taipei, Taiwan.
- Taku Kudo, Taiyaki Komatsu, Toshiyuki Hanaoka, Jun Mukai, and Yusuke Tabata. 2011. Mozc: A statistical kana-kanji conversion system (in japanese). In *Proceedings of Japan Natural Language Processing*, pages 948–951.
- Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. 2009. Collocation extraction using monolingual word alignment method. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 487–495, Singapore, August. Association for Computational Linguistics.
- Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. 2010. Improving statistical machine translation with monolingual collocation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 825–833, Uppsala, Sweden, July. Association for Computational Linguistics.
- Kikuo Maekawa. 2008. Compilation of the kotonoha-bccwj corpus (in japanese). *Nihongo no kenkyu (Studies in Japanese)*, 4:82–95.
- Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, May.
- Shinsuke Mori, Masatoshi Tsuchiya, Osamu Yamaji, and Makoto Nagao. 1999. Kana-kanji conversion by a stochastic model (in japanese). *Journal of Information Processing Society of Japan*, 40(7).
- Naoaki Okazaki and Sophia Ananiadou. 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(22):3089–3095.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.
- Juergen Schroeter, Alistair Conkie, Ann Syrdal, Mark Beutnagel, Matthias Jilka, Volker Strom, Jun Kim, Hong goo Kang, and David Kapilow. 2002. A perspective on the next challenges for tts research. In *Proceedings of 2002 IEEE Workshop on Speech Synthesis*.
- Kosho Shudo, Akira Kurahone, and Toshifumi Tanabe. 2011. A comprehensive dictionary of multi-word expressions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 161–170, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Hisami Suzuki and Jianfeng Gao. 2005. Microsoft research ime corpus. Technical Report MSR-TR-2005-168, Microsoft Research.
- Xianchao Wu, Naoaki Okazaki, and Jun’ichi Tsujii. 2009. Semi-supervised lexicon mining from parenthetical expressions in monolingual web pages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 424–432, Boulder, Colorado, June. Association for Computational Linguistics.

# A Factoid Question Answering System Using Answer Pattern Matching

**Nagehan Pala Er**

Department of Computer Engineering,  
Bilkent University, Ankara, Turkey  
nagehan@cs.bilkent.edu.tr

**Ilyas Cicekli**

Department of Computer Engineering,  
Hacettepe University, Ankara, Turkey  
ilyas@cs.hacettepe.edu.tr

## Abstract

In this paper, we describe a Turkish factoid QA system which uses surface level patterns called answer patterns in order to extract the answers from the documents that are retrieved from the web. The answer patterns are learned using five different answer pattern extraction methods with the help of the web. Our novel approach to extract named entity tagged answer patterns and our new confidence factor assignment approach have an important role in the successful performance of our QA system. We also describe a novel query expansion technique to improve the performance. The evaluation results show that the named entity tagging in answer patterns and the query expansion leads to significant performance improvement. The scores of our QA system are comparable with the results of the best factoid QA systems in the literature.

## 1 Introduction

Question answering is the task of returning a particular piece of information in response to a natural language question. The aim of a question answering system is to present the required information directly, instead of documents containing potentially relevant information. Questions can be divided into five categories (Modolvan et al., 2002; Schone et al., 2005): factoid questions, list questions, definition questions, complex questions, and speculative questions. A factoid question has exactly one correct answer which can be extracted from short text segments. The difficulty level of factoid questions is lower than the other categories. In this paper, we present a Turkish factoid question answering system which retrieves the documents that contain answers, and extracts the answers from these retrieved documents with the help of a set of learned answer extraction patterns. List, defini-

tion, complex, and speculative questions are out of the scope of this paper.

At TREC-10 QA track (Voorhees, 2001), most of the question answering systems used Natural Language Processing (NLP) tools such as a natural language parser and WordNet (Fellbaum, 1998). However, the best performing system at TREC-10 QA track used only an extensive list of surface patterns (Soubbotin and Soubbotin, 2001). Therefore we have decided to investigate their potential for Turkish factoid question answering. Our factoid question answering system learns answer patterns that are surface level patterns, and it uses them in the extraction of the answers of new questions. Our answer patterns are learned from the web using machine learning approaches. In addition to the creation of raw string answer patterns, we tried different methods for answer pattern creation such as stemming and named entity tagging. Our novel answer pattern creation method using named entity tagging produces most successful results.

One of the important issues in the factoid question answering is the answer ranking. The correct answer for a question should be in the top of the produced answer list by a QA system. The learned answer patterns in our QA system are associated with confidence factors, and their confidence factors indicate their precision values for the training set. The confidence factors of the rules that extracted the answers are also used to rank the answers and this approach produces very good results.

The question answering systems that extract the answers from the retrieved web pages should be able to retrieve the web pages that contain the answers. These QA systems form a search engine query and submit this query to retrieve the related web pages containing the answers. The retrieved web pages may or may not contain the answers. The QA systems can only consider the first retrieved web pages in order to extract an-

swers from them, and the QA systems have a bigger chance to extract the correct answers if the first retrieved web pages contain the answers. In order to increase the chance that the first retrieved web pages contain the possible answers, we apply a novel query expansion approach using answer patterns. Our evaluation results indicate that our query expansion approach improves the performance of our factoid question answering system.

The factoid question answering system described here is the first successful Turkish factoid question answering system. Our new confidence factor assignment approach to the learned answer patterns has an important role in the success of our factoid QA system. The contributions of our paper also include the introduction of a novel query expansion approach and the creation of named entity tagged answer patterns. The performance results of our factoid question answering system are competitive with the results of the state of art QA systems in the literature.

The rest of the paper is organized as follows. Section 2 discusses our answer pattern extraction methods and confidence factor assignments to the extracted answer patterns. In Section 3, we describe the question answering phase of our QA system. Section 4 presents the detailed discussions about the evaluation results. Section 5 contains concluding remarks.

## 2 Answer Pattern Extraction

In the learning phase of our question answering system, a set of answer patterns are inferred for each question type using the training set of that question type and the web. For each question type, we prepared a set of training examples which consists of question and answer phrase pairs. A query is formed as a conjunction of question and answer phrases in each training example. This query is used to retrieve top documents (*DocSet1*) containing the question and answer phrases. The retrieved documents are used in the extraction of answer patterns without confidence factors. For each training example, we also form a query which only consists of the question phrase. The retrieved documents (*DocSet2*) using this query may or may not contain answer phrases. Two document sets are used in the calculation of the confidence factors of the learned answer patterns.

Although the retrieved documents in *DocSet1* contain both question and answer phrases, question and answer phrases may not appear together

in a sentence of a document. In order to determine answer pattern strings, the sentences that contain the question and the answer phrases together are selected from documents, and answer pattern strings are extracted from these sentences. An answer pattern string is a substring that starts with the answer phrase and ends with the question phrase, or starts with the question phrase and ends with the answer phrase. In addition, we extract an answer pattern string with a boundary word in order to determine the boundary of the answer phrase.

After an answer pattern string is extracted, we apply five different methods in order to learn answer patterns: Raw String (*Raw*), Raw String with Answer Type (*RawAT*), Stemmed String (*Stemmed*), Stemmed String with Answer Type (*StemmedAT*), and Named Entity Tagged String (*NETagged*). Raw string methods learn more specific rules than Stemmed string methods, and *NETagged* method learns more general rules.

In order to extract a raw string answer pattern using our *Raw* method, question phrase and answer phrase in an answer pattern string are replaced with appropriate variables QP and AP. QP is replaced with the given question phrase during question answering, and AP is bound to the answer phrase of the question if the pattern matches. The length of the found answer phrase is determined by the boundary word if the answer pattern contains a boundary word. Otherwise, a fixed size is used as its length.

There can be many strings that can match with an answer pattern that is learned using *Raw* method. One reason for this is that there is no type checking for the string to which AP binds. As long as the pattern matches, AP binds with a string. Our *RawAT* method associates AP variables with answer types. An answer type is a named entity type that is determined by our Turkish named entity tagger. During question answering, the found answer phrase is checked by our named entity tagger in order to make sure that it satisfies the type restriction. For this reason, an answer pattern with an answer type is more specific than the corresponding answer pattern without a type.

Answer patterns obtained by raw string methods contain surface level words, and they have to match exactly with words in extracted strings. Stemmed string methods replace words with their stems in answer patterns. In order to match a string with a stemmed answer pattern, all its words are stemmed first and its stemmed version matches with the stemmed answer pattern to ex-

tract the answer. The extracted answer patterns can be still more specific since they may contain specific words. *NETagged* method can further generalize answer patterns by replacing all named entities in the string by typed variables.

After all answer patterns are extracted from the training set, the confidence factors are assigned to these extracted answer patterns. A confidence factor of an answer pattern indicates its accuracy in the training set. In question answering phase, we use only the answer patterns whose confidence factors are above a certain threshold. From two document sets (*DocSet1* and *DocSet2*), the sentences containing the question phrase are collected as a training set for confidence factor assignment. The confidence factor of an answer pattern is the proportion of correct results to all results extracted by that pattern.

### 3 QA Using Answer Patterns

Our base question answering module uses the given question phrase as a search engine query. Using Bing web search engine top documents containing the given question phrase are retrieved. In these documents, the sentences containing the question phrase are extracted. The question phrases in the retrieved sentences are replaced by QP, and these sentences are used in the answer processing phase.

In the answer processing phase, the answer patterns of the given question type are applied to the selected sentences in order to extract answer phrases. The preprocessing of the sentences may be required depending on the method of the used answer pattern. If the applied method is a raw string method, there is no need for the preprocessing of the sentence, and the raw string answer pattern is directly applied to the sentence. If the answer pattern is a stemmed string answer pattern, all the words are stemmed first, and the answer pattern is applied to the stemmed version of the sentence. If the answer pattern is a NE tagged answer pattern, the sentence is analyzed by the named entity tagger in order to determine all named entities in the sentence, and the answer pattern is applied to the named entity tagged version of the sentence.

If the applied answer pattern matches the sentence, an answer phrase is extracted as a result. If the answer phrase in the applied answer pattern is named entity tagged, the extracted answer phrase must also satisfy conditions of that named entity. The confidence value of an extracted answer is the confidence factor of the matched an-

swer pattern. The top ranked answer is returned as the result of the question.

Our base QA algorithm creates a search engine query and that query only contains the given question phrase. The retrieved documents may be insufficient to extract the correct answer because the query is too general and the retrieved documents may not contain the answer. We want to retrieve documents that contain many sentences holding the question phrase and answer phrase together. Thus, there is a bigger chance that our answer patterns match those sentences, and the correct answer can be extracted. In order to retrieve the documents that are more likely to contain the answer, we use a query expansion approach. The answer patterns with high confidence factors are used to expand the query, so that the more related documents can be retrieved.

### 4 Evaluation Results

In order to evaluate the performance of our system, we prepared a training set and a test set and they do not contain any common item. Each of them contains 15 question-answer phrase pairs from seven different question types (Author, Capital, DateOfBirth, DateOfDeath, LanguageOfCountry, PlaceOfBirth, PlaceOfDeath). Since we obtained our best results, when we use the answer patterns higher than 0.75 confidence factors, we only used these answer patterns for evaluations. The answer patterns are tested with the question-answer phrase pairs in the test set.

We used four standard evaluation metrics: Precision, Recall, Fmeasure and MRR. Precision is the proportion of the number of correct answers to the number of returned answers, and Recall is the proportion of the number of correct answers to the number of test questions. Fmeasure is the harmonic mean of Precision and Recall. Mean Reciprocal Rank (MRR) considers the rank of the first correct answer in the list of possible answers (Voorhees, 2001).

	<i>MRR</i>	<i>Recall</i>	<i>Precision</i>	<i>Fmeas</i>
<i>Raw</i>	0.28	0.24	0.57	0.34
<i>RawAT</i>	0.31	0.30	0.86	0.44
<i>Stemmed</i>	0.29	0.26	0.57	0.36
<i>StemmedAT</i>	0.30	0.29	0.88	0.44
<i>NETagged</i>	0.45	0.45	0.94	0.61
<i>AllWithNE</i>	0.58	0.56	0.86	0.68

Table 1. Evaluation results

We evaluated each of our five methods separately and their best combination. The evaluation results are given in Table 1. The results in the

columns 2-5 of Table 1 are the average values of the results of the seven question types. The rows 2-6 give the results for individual methods and the last row gives the results of their best combination *AllWithNE* which contains the answer patterns that are learned from methods *RawAT*, *StemmedAT* and *NETagged*. According to the results, our best method is *NETagged* method which accomplishes the best scores for all four evaluation metrics. These results indicate that the usage of named entity tagged string answer patterns increases the performance. The results indicate that the effect of stemming is not as good as expected. The usage of answer types blocks the extraction of most of the incorrect answers.

In our query expansion method, we use the words appearing in the high confidence answer patterns. One way to test the effectiveness of our query expansion mechanism is to measure the change in the number of sentences containing both the question phrase and the answer phrase in the retrieved documents. According to our results, the number of such sentences is increased from 3227 to 6647 when the query expansion is employed. This means that our answer patterns have almost twice the chance to extract answers using query expansion. We applied our answer patterns in our best combination *AllWithNE* to the documents returned as a result of the query expansion. The highest increase (29%) occurred in Recall result because the answers of more questions are retrieved as a result of the query expansion. Precision result is also improved from 0.86 to 0.94 (9% increase). The increase in MMR result is 26% percent and the increase in Fmeasure result is 20%. As a conclusion, the query expansion is a useful tool to improve the performance since it leads to increases in all measures.

<i>QA System</i>	<i>MRR</i>
TREC-8 (max,avg,min)	0.66, 0.25, 0.02
TREC-10 (max,avg,min)	0.68, 0.39, 0.27
Ephyra	0.40
Ravichandran and Hovy's QA sys.	0.57
BayBilmis	0.31
Our Best without query expansion	0.62
Our Best with query expansion	0.73

Table 2. Comparisons of QA systems

Although it is difficult to directly compare the results of our QA system with the published results of other factoid question answering systems, we still discuss the MRR results of our QA system and other factoid question answering

systems. Table 2 compares our best MRR results with the MRR results of the other systems (Voorhees, 1998; Voorhees, 2001; Schlaefter and Gieselmann, 2006; Ravichandran and Hovy, 2001; Amasyali and Diri, 2005). Although these scores may not give fair comparisons, they still show that our QA system is competitive to the best factoid QA systems.

## 5 Conclusion and future work

The answer pattern matching technique has been used successfully for English Factoid QA (Ravichandran and Hovy, 2001; Schlaefter and Gieselmann, 2006; Soubbotin and Soubbotin, 2001), we therefore decided to apply various answer pattern extraction methods for Turkish factoid QA. These methods are compared according to MRR, Fmeasure, Recall and Precision scores. The scores of stemmed string methods are slightly better than the scores of raw string methods, so stemming slightly improves the performance of the system. The scores of *RawAT* and *StemmedAT* methods are better than the scores of *Raw* and *Stemmed* methods, so checking the answer type improves the performance of the system significantly. *NETagged* method has the best scores. So, replacing words with their named entity tags improves the performance.

We have also implemented a novel query expansion approach using answer patterns. We use the most reliable raw string answer patterns to extend queries. The number of sentences containing the answer phrase increases when the query expansion is applied. The performance scores increase significantly when the query expansion is applied.

The question answering system described in this paper is the first successful Turkish factoid question answering system. The evaluation results indicate that the performance of our QA system is comparable with the performances of the state of the art factoid question answering systems.

Investigating the potential of more generic answer patterns is left as a future work. *Stemmed*, *StemmedAT* and *NETagged* methods extract more generic answer patterns compared to *Raw* and *RawAT* methods and they achieve better results. More generic answer patterns can be extracted by using linguistic techniques such as phrase chunking and morphological analysis. We believe that combining different answer processing techniques can improve the performance of the QA system significantly.

## References

- M.F. Amasyalı and B. Diri, Bir soru cevaplama sistemi: Baybilmiş, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 2005 (in Turkish).
- C. Fellbaum, *WordNet: An Electronic Lexical Database*, The MIT Press, (1998).
- D. Modolvan, M. Pasca, S. Harabagiu and M. Surdeanu, Performance issues and error analysis in an open-domain question answering system, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)* (2002).
- R. Ravichandran and E. Hovy, Learning surface text patterns for a question answering system, *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (2001).
- N. Schlaefer and P. Gieselmann, A pattern learning approach to question answering within the Ephyra framework, *Proceedings of the 9th International Conference on Text, Speech and Dialogue* (2006).
- P. Schone, G. Ciary, R. Cutts, J. Mayfield and T. Smith, QACTIS-based question answering at TREC-2005, *Proceedings of the 14th Text Retrieval Conference* (2005).
- M. Soubbotin and S. Soubbotin, Patterns of potential answer expressions as clues to the right answer, *Proceedings of the 10th Text Retrieval Conference* (2001).
- E.M. Voorhees, The TREC-8 question answering track report, *Proceedings of the 8th Text Retrieval Conference* (1999).
- E.M. Voorhees, Overview of the TREC 2001 question answering track, *Proceedings of the 10th Text Retrieval Conference* (2001).

# Chinese Short Text Classification Based on Domain Knowledge

**Xiao Feng**

Institute of Automation  
Chinese Academy of Science  
xiao.feng@ia.ac.cn

**Yang Shen**

State Administration for In-  
dustry & Commerce of the  
People's Republic of China  
shenyang@saic.gov.cn

**Chengyong Liu**

Information Center of Gen-  
eral Administration of Press  
and Publication of PR China  
liucy\_gapp@sina.com

**Wei Liang**

Institute of Automation  
Chinese Academy of Science  
wei.liang@ia.ac.cn

**Shuwu Zhang**

Institute of Automation  
Chinese Academy of Science  
shuwu.zhang@ia.ac.cn

## Abstract

People are generating more and more short texts. There is an urgent demand to classify short texts into different domains. Due to the shortness and sparseness of short texts, conventional methods based on Vector Space Model (VSM) have limitations. To tackle the data scarcity problem, we propose a new model to directly measure the correlation between a short text instance and a domain instead of representing short texts as vectors of weights. We firstly draw domain knowledge for each user-defined domain using an external corpus of longer documents. Secondly, the correlation is calculated by measuring the proportion of the overlapping part of the instance and the domain knowledge. Finally, if the correlation is greater than a threshold, the instance will be classified into the domain. Experimental results show that the classifier based on the proposed model outperforms the state-of-the-art baselines based on VSM.

## 1 Introduction

In recent years, web services are generating more and more short texts including micro-blogs, customer reviews, chat messages and so on. However, a user is often only interested in very small part of these data. There is an urgent demand to classify incoming short texts into different domains, so that users are not overwhelmed by the raw data. As short texts do not provide sufficient word occurrences (i.e., the length of a micro-blog is limited to 140 characters), conventional text classifiers often cannot achieve high accuracy,

especially when the number of training examples is small.

Vector Space Model (VSM) is a very popular document representation model, where each document is represented as a vector of weights. Text classification methods based on VSM perform well when processing documents in regular length (Berry and Michael, 2004). But, the sparsity of VSM will reduce the classification accuracy when processing short texts.

There have been several studies that attempted to solve the problem of data sparseness in VSM. One way is to select more useful features using additional semantics from Wikipedia (Banerjee *et al.*, 2007), WordNet (Hu *et al.*, 2009) or HowNet (Liu *et al.*, 2010). Another way is to expand the coverage of classifier by using background knowledge drawn from much longer external data sources. Zelikovitz and Hirsh (2000) utilized a corpus of unlabeled longer documents as a “bridge”, to connect the test example with training examples. Phan *et al.* (2008) and Chen *et al.* (2011) integrated the original short text with hidden topics discovered from external large-scale data collections to add more meta-information. These researches have shown positive improvement by enriching the representation of feature vectors, but a disadvantage is the high computational complexity.

In this paper, we try to solve the sparse problem from another direction with lower computational complexity. We propose a new model to directly measure the correlation between a short text instance and a domain, using domain knowledge drawn from a labeled external corpus of related longer documents. We performed a careful evaluation for our model on micro-blog



classification task, and achieved consistent improvements over two baselines.

The overall framework of our approach is shown in Figure 1. We firstly draw domain knowledge for each user-defined domain using the external corpus. Secondly, the correlation between a short text instance and a domain is calculated by measuring the proportion of the overlapping part of this instance and the domain knowledge of this domain. Finally, if the correlation is greater than a threshold, the instance will be classified into the domain. The main advantages of our approach include the following points:

- Good generalization performance: domain knowledge learned from longer documents can cover lots of terms that do not exist in a small labeled training set.
- Easy to implement: No need to construct VSM to train classifiers. All we need to prepare is the domain knowledge.

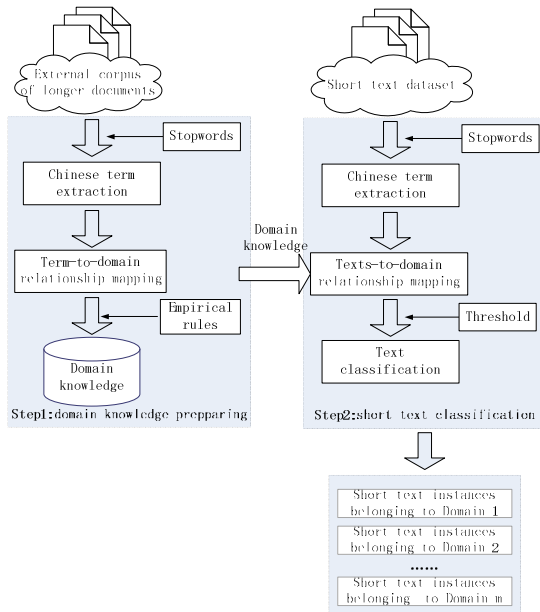


Figure 1. The framework of our approach

## 2 Our Approach

### 2.1 Domain knowledge preparing

To expand the coverage of our model, we utilize a labeled external corpus of longer documents to draw domain knowledge. Text documents in this corpus have been classified into user-defined domains. There are two main conditions that should be followed to choose an appropriate external corpus. First, the coverage of vocabulary should be sufficient. Second, the user-defined

domains should be consistent with the classification problem. In fact, the second condition will not be a problem, because a large number of web documents belonging to different domains can be crawled from portal sites such as Sina<sup>1</sup> and Sohu<sup>2</sup>.

After Chinese term extraction and removing stop words, we obtain an initial term list appearing in the corpus, denoted by  $T = \{t_1, t_2, \dots, t_n\}$ .

The aim of term-to-domain relationship mapping is to select  $K$ -number of most related terms for each domain from  $T$ . The set of selected terms is regarded as the domain knowledge of a domain. How terms are related to each domain is measured by applying Chi-square statistical term-to-domain independency measurement. The measurement is based on the co-occurrence frequencies of a term and a domain. We firstly assume that the term and the domain are statistically independent, and then compare the observed frequency and the expected frequency.

Let  $t_i (i = 1, 2, \dots, n)$  be a term in the initial term list  $T$ , and  $d_j (j = 1, 2, \dots, m)$  be a domain in the user-defined domain list  $D = \{d_1, d_2, \dots, d_m\}$ . The expected frequency is defined as:

$$E_{e_i e_c} = \frac{\sum_{p \in \{0,1\}} O_{pe_c} \sum_{q \in \{0,1\}} O_{e_i q}}{N}, e_i \in \{0,1\}, e_c \in \{0,1\} \quad (1)$$

where  $N = O_{11} + O_{01} + O_{10} + O_{00}$ ,  $O_{11}$  denotes the observed frequency of documents which contain  $t_i$  and belong to  $d_j$ ,  $O_{01}$  denotes the observed frequency of documents which do not contain  $t_i$  but belong to  $d_j$ ,  $O_{10}$  denotes the observed frequency of documents which contain  $t_i$  but not belong to  $d_j$ , and  $O_{00}$  denotes the observed frequency of documents that neither contain  $t_i$  nor belong to  $d_j$ .

The Chi value for  $t_i$  and  $d_j$  is defined as:

$$\chi^2(t_i, d_j) = \sum_{e_i \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(O_{e_i e_c} - E_{e_i e_c})^2}{E_{e_i e_c}} \quad (2)$$

Note that the greater the Chi value is, the closer the relationship between  $t_i$  and  $d_j$  is.

Let  $\overline{DK} = \{dk_1, dk_2, \dots, dk_m\}$  be the domain knowledge vector,  $dk_j$  be the domain knowledge of domain  $d_j$ ,  $\overline{tr}_i$  be the Chi value vector of term  $t_i$ , and  $\overline{dr}_j$  be the Chi value vector of do-

<sup>1</sup> <http://www.sina.com.cn/>

<sup>2</sup> <http://www.sohu.com/>

main  $d_j$ . The algorithm of term-to-domain relationship mapping includes three main steps:

**Step1:** For each term  $t_i$ , construct its Chi value vector  $\bar{tr}_i = \{\chi^2(t_i, d_1), \chi^2(t_i, d_2), \dots, \chi^2(t_i, d_m)\}$ .

**Step2:** For each  $\bar{tr}_i$ , find its largest item  $\chi^2(t_i, d_j)$  and put it into  $\bar{dr}_j$ .

**Step3:** Sort items in  $\bar{dr}_j$  in descending order. Select the corresponding terms of the first  $K$  - number of items, and put them in  $dk_j$ . All terms in  $dk_j$  are arranged in descending order under its Chi value.

## 2.2 Short text classification

In this section, we introduce an intuitive model to directly relate each short text instance to one or more specific domains. How a short text instance, denoted by  $g$ , is related to a domain  $d_j$  can be measured based on the correlation between them. The correlation is calculated by measuring the proportion of the overlapping part of  $g$  and the domain knowledge  $dk_j$  (Liu *et al*, 2012), see Figure 2.

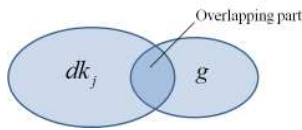


Figure 2. The overlapping part of a short text instance and the domain knowledge

To measure the proportion of the overlapping part, we need to compute the score of domain knowledge  $dk_j$  and the score of the overlapping part of  $g$  and  $dk_j$ , and then normalize the latter by the former. Moreover, we introduce a weight, denoted by  $w_{jk}$ , to indicate the importance of a term in  $dk_j$ , when calculating the scores. As the Chi values of terms in different domain knowledge vary greatly, we define the weight of a term based on its ordering position in the domain knowledge. The weight is defined as:

$$w_{jk} = \frac{K+1-k}{K}, k=1,2,\dots,K \quad (3)$$

where  $k$  is the order of the term in  $dk_j$ , and  $K$  is the number of terms in  $dk_j$ .

Finally, the correlation between  $g$  and  $d_j$  is calculated based on scores as:

$$score_{dk_j} = \frac{1}{L} \sum_{k=1}^K w_{jk} \quad (4)$$

$$score_{overlapping} = \sum_{k=1}^K w_{jk} \times \frac{tf(t_{jk})}{len(g)} \quad (5)$$

$$correl(g, d_j) = \frac{score_{overlapping}}{score_{dk_j}} \quad (6)$$

where  $L$  is the average length of documents in the external corpus ( i.e. the average size of the term lists of documents),  $tf(t_{jk})$  is the frequency of term  $t_{jk}$  appearing in  $g$ , and  $len(g)$  is the length of  $g$ .

If  $correl(g, d_j)$  is greater than a threshold  $\delta$ ,  $g$  will be classified into  $d_j$ . The optimized value of  $\delta$  can be obtained by cross-validation.

## 3 Experiments and results

### 3.1 Data Sets

We collect short texts from Sina micro-blog<sup>3</sup>, and use an open corpus<sup>4</sup> collected by Sogou Lab from the Internet as the external corpus.

**External Corpus** Documents belong to 8 domains: Finance, IT, Health, Sports, Tour, Education, Film&TV, and Military. Each domain contains 600 documents. The vocabulary is 69909 terms. The average length of documents is 403 terms.

**Micro-blog Dataset** We manually choose training samples and test samples for each user-defined domain. Samples in the training set and the test set are totally exclusive. The average length of micro-blogs is 31 terms. There is a noise set in the test set containing micro-blogs which do not belong to any user-defined domain, see Table 1.

Domain	#Train data	#Test data
Finance	600	300
IT	600	300
Health	600	300
Sports	600	300
Tour	600	300
Education	300	150
Film&TV	600	300
Military	300	150
<b>Noise set</b>	0	10,000
<b>Total</b>	4200	12100

Table 1. Description of the micro-blog dataset

<sup>3</sup> <http://t.sina.com.cn>

<sup>4</sup> <http://www.sogou.com/labs/dl/c.html>

### 3.2 Measurement

We adopt the F1-measure as our performance criterion to balance the influence between precision and recall.

$$precision = \frac{TP}{TP + FP} \quad (7)$$

$$recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1\text{-measure} = \frac{2 \times precision \times recall}{precision + recall} \quad (9)$$

where  $TP$  denotes the numbers of relevant samples classified as relevant,  $FN$  denotes the numbers of relevant samples classified as irrelevant, and  $FP$  denotes the number of irrelevant samples classified as relevant.

### 3.3 Experiment results and analysis

In order to obtain the optimized value of  $\delta$ , we randomly divided the training set into five equal partitions and performed 5-fold cross-validation. In Table 2, we can find that our classifier achieves the highest F1-measure when  $\delta = 0.2$  and  $K = 500$ . Thus in all following experiments, we employ  $\delta = 0.2$ .

$\delta$ $K$	0.1	0.2	0.3	0.4	0.5
100	0.8201	0.8847	0.9110	0.9189	0.9125
200	0.8635	0.9274	0.9476	0.9421	0.8821
300	0.8934	0.9506	0.9447	0.9235	0.7203
400	0.9187	0.9519	0.9417	0.7286	0.4537
500	0.9388	<b>0.9554</b>	0.8494	0.5463	0.2293
600	0.9453	0.9523	0.7545	0.4372	0.1120
700	0.9482	0.8938	0.6842	0.3213	0.0503

Table 2. 5-fold cross-validation on training set

The next experiment is to compare our method with two baselines based on VSM of TFIDF weights on the test set. Both the baselines are composed of 8 SVM (Support Vector Machine) classifiers (one for each domain to decide whether a test sample belongs to this domain). One of them uses terms in domain knowledge drawn from the external corpus as features to enrich the representation of VSM (“VSM with E” for short). The other one only uses terms in domain knowledge drawn from the training set as features to construct VSM (“VSM without E” for short). We use RBF kernel and optimized parameters which are chosen by grid-search in LIBSVM<sup>5</sup> to train SVM classifiers.

<sup>5</sup> <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

	VSM without E	VSM with E	Our Method
Optimized $K$	700	500	500
Optimized F1-measure	0.8965	0.9285	<b>0.9520</b>

Table 3. The overall optimized results of our method and VSM-based methods

Table 3 shows the overall optimized result of each method with its optimized  $K$  on the test set, and Figure 3 shows the optimized result of each domain in more details. We can find that our method achieves the highest overall F1-measure, and achieves 5.7%, 1.7%, 3.0%, 1.9%, 3.5%, 0.06%, 2.9% and 1.4% improvements over VSM with E for Finance, IT, Health, Sports, Tour, Education, Film&TV, and Military respectively.

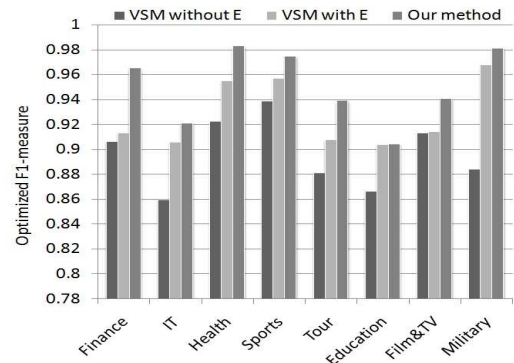


Figure 3. Optimized results of each domain in more details

## 4 Conclusion

In this paper, we propose a new model based on domain knowledge to solve the data sparsity problem in short text classification. We validate through experiments that classifier based on our model outperforms classifiers based on VSM. In the future work, we will try to combine the external knowledge and the training set to further improve the performance of short text classification.

### Acknowledgments

The work has been supported by the National Key Technology R&D Program of China under Grant No. 2011BAH16B02, 2013BAH61F01 and 2012BAH88F03.

## References

- Banerjee S, Ramanathan K, and Gupta A. 2007. *Clustering Short Texts Using Wikipedia*. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007: 787-788.
- Hu X, Sun N, Zhang C, and Chua T. 2009. *Exploiting internal and external semantics for the clustering of short texts using world knowledge*. Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009: 919-928.
- Zelikovitz S and Hirsh H. 2000. *Improving short text classification using unlabeled background knowledge to assess document similarity*. Proceedings of the Seventeenth International Conference on Machine Learning. 2000: 1183-1190.
- Phan X, Nguyen L, and Horiguchi S. 2008. *Learning to classify short and sparse text & web with hidden topics from large-scale data collections*. Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 91-100.
- Chen M, Jin X, and Shen D. 2011. *Short text classification improved by learning multi-granularity topics*. Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three. AAAI Press, 2011: 1776-1781.
- Liu Z, Yu W, Chen W, et al. 2010. *Short text feature selection for micro-blog mining*. Computational Intelligence and Software Engineering (CiSE), 2010 International Conference. IEEE, 2010: 1-4.
- Berry, and Michael. 2004. *Survey of Text Mining I: Clustering, Classification, and Retrieval*. volume 1. Springer-Verlag New York Incorporated, 2004.
- Liu J N K, He Y, Lim E H Y, et al. 2012. *Domain ontology graph model and its application in Chinese text classification*. Neural Computing and Applications, 2012: 1-20.

# Applying Graph-based Keyword Extraction to Document Retrieval

Youngsam Kim<sup>1</sup> Munhyong Kim<sup>1</sup> Andrew Cattle<sup>1</sup> Julia Otmakhova<sup>1</sup>  
Suzi Park<sup>1</sup> Hyopil Shin<sup>1</sup>

<sup>1</sup>Seoul National University/ Gwanak-1, Gwanak-ro, Gwanak-gu, Seoul, South Korea  
youngsam@gmail.com, likerainsun@gmail.com, acattle@gmail.com,  
julia.nixie@gmail.com, mam3b@snu.ac.kr, hpshin@snu.ac.kr

## Abstract

This paper proposes a keyword extraction process, based on the PageRank algorithm, to reduce noise of input data for measuring semantic similarity. This paper will introduce several features related to implementation and discuss their effects. It will also discuss experimental results which showed significantly improved document retrieval performance with this extraction process in place.

## 1 Introduction

To date, the most popular and well known approach to calculating semantic similarity of text documents has been utilizing Vector Space Models (VSM). The key idea of VSM is to map each document in a corpus or collection into a vector of a vector space, and interpret the distance between the query document's vector and the other texts' vectors as their degree of semantic relatedness (Salton, 1971; Salton et al., 1975; Turney and Pantel, 2010).

The VSM evolved from the SMART information retrieval system (Salton, 1971) and SMART pioneered many important terms and concepts that were adopted by modern search engines (Turney and Pantel, 2010). Many search engines are reported to use VSM to calculate the similarity between a query and a document (Manning et al., 2008).

A common problem of VSM is that documents often contain words with high frequency but little semantic significance. VSM usually deal with this obstacle using tf-idf weighting and singular value decomposition techniques (Turney and Pantel, 2010).

In order to improve retrieval systems that use the models, we suggest that employing keyword

extraction on a per document basis to help reduce the noise inherent in large texts. For this purpose, PageRank, a graph-based ranking algorithm, was used in this study. Graph-based analysis techniques represent a document or text as a graph consisting of nodes (terms or phrases) and edges (pre-defined relations). Along with Brin and Page (1998)'s PageRank, various modifications and other graph-based algorithms have been introduced and proved their usefulness in various natural language processing tasks (Erkan and Radev, 2004; Kurland and Lee, 2006; Mihalcea et al., 2004; Wang et al., 2007; Widdows and Dorow, 2002).

Graph-based extraction systems showed better performance over frequency-based systems on multiple-theme documents (Grineva et al., 2009). In this study, it was assumed that applications would benefit from being able to select important words from documents using the extraction system; not just for keyword extraction tasks, but also for any complex system that needs its input-data to be noise-reduced for future processes.

## 2 Theoretical Background

In this paper, typical core parts of VSM are applied to measure semantic similarity over documents. Therefore, instead of using raw frequencies tf-idf weighting is adopted and length normalization is performed on both queries and target documents (Salton and Buckley, 1988; Buckley, 2005). In addition, the traditional cosine similarity is used to calculate closeness scores between pseudo documents (queries) and documents (following Buckley, 2005). However, the vector space dimensionality reduction phase has been omitted to simplify the experiment process.

### 2.1 Representing Text as Graphs

Before applying the Graph-based approach, several preprocessing stages had to be implemented.

First of all, the words in the text were identified as vertices of the graph. In this study, only unigrams were considered as node candidates. These unigrams were then POS tagged and passed through a syntactic filter, which only allowed a particular subset of POS tags. Various syntactic filters were experimented with including nouns, verbs, and adjectives but the best results were obtained when only nouns were used.

## 2.2 Defining Relationship Between Vertices

The relationship between vertices was chosen to be a co-occurrence relationship. Two nouns of the text would be connected if they both occurred within a window of  $N$  pre-fixed words.  $N$  can be any integer from 2 to 10, but the number of vertices,  $V$  is always equal to or less than  $N$  because the words in the window must bear a relevant POS tag from a predefined set.

In English it is easy to determine which words are within a specific window since each word split by spaces usually corresponds to one POS tag. However, unlike English, Korean is an agglutinative language and most words consist of more than one morpheme, each with their own part of speech. The example below, (1) demonstrates this fact.

- (1) 그 여자가 학교에 갔다.  
 Ku nyeca-ka hakkyo-ey ka-ss-ta  
 The(ku) woman(nyeca)-Normative(ka)  
 school(hakyo)-Locative(ey) go(ka)-  
 PST(ss)-FinalSuffix(ta)  
 ‘The woman went to school’

If the sentence above is POS tagged, the number of tagged members would be eight, three tags more than the English equivalent. What this means is that the average distance between particular POS tagged items in Korean sentences is longer than in English sentences. Presumably, this would lengthen optimal window size in Korean when compared to English. According to the related result of Mihalcea et al. (2004), the best performance was achieved with the window of 2. In our case, it is natural to assume the span of the window would be wider.

However, it is also possible to consider the segments divided by whitespaces as the candidate nodes for the graph, and perform POS tagging after the separation. In this way, we can disregard the distance between any lexeme and functional prefix/suffix attached to it. Under this scheme the normative case ‘ka’ in the middle is

ignored and thus the words ‘woman’ and the ‘school’ in (1) have no distance between them.

This second approach is very similar to the way in which English text is translated as a graph, but it disregards information gained from grammatical relations between nouns and functional prefixes/suffixes glued to them. These two approaches were both experimented with and their results are compared in Section 4.1.

## 3 Experiment

### 3.1 System Framework

The components explained above were implemented in an integrated system, including text pre-processing, POS tagging, keyword extraction, term weighting, and finally calculating semantic similarity. The goal of this system was to retrieve semantically related documents using query documents from the collected corpus.

The general workflow of the system is presented in Fig. 1. The system is designed for easy addition or removal of any of the intermediate stages or processes for experimental purposes. Such configuration changes constituted the different experiment conditions of this research.

Text pre-processing indicates deletion of any special characters, emoticons, and foreign words to allow the sanitized text to be parsed safely during part of speech analysis. And the POS tagger assigns each morpheme one of 22 tag sets and the words given the tag of ‘noun’, excluding pronouns, are passed for later processing.

To establish the stop-word list, three reviewers examined all the nouns extracted from the 800 documents that were collected for this study and selected 192 lexical items manually only if at least two of the reviewers voted for the same word to be on the list.

The Graph-based Keyword Extraction, Frequency Counting, and tf-idf Weighting modules may vary or be absent across various experiment conditions (This is denoted by the dotted boxes in Fig. 1). Keyword extraction stage always follows after Graph analysis because it is the process for sorting and choosing the adequate number of keywords. If the graph analysis is not performed, there can be no keyword extraction.

The Graph-based Keywords Extraction and Frequency Counting are mutually exclusive and only one method is chosen for each experiment condition.

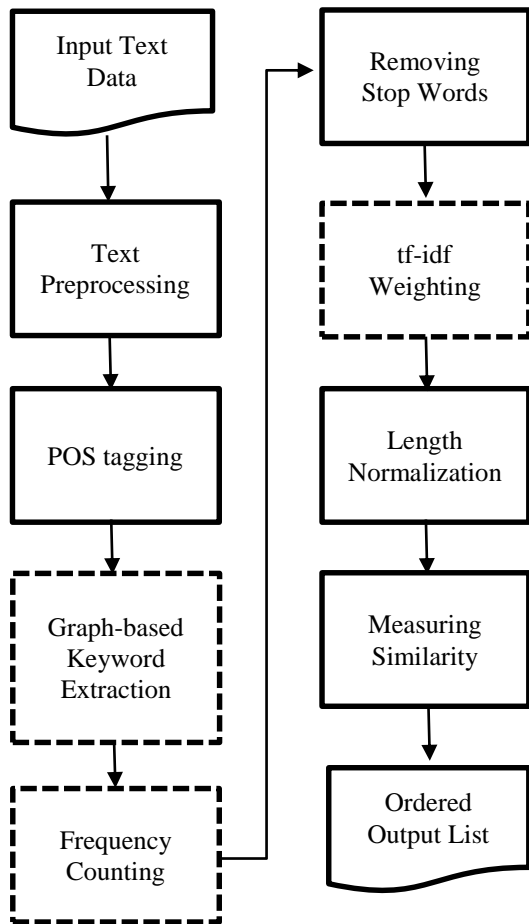


Figure 1. System workflow

### 3.2 Experiment Conditions

Three conditions in total were examined using the system. The first model, using graph-based ranking algorithm, adopted the tf-idf weighting method but omits the Frequency Counting stage. As such, the tf-idf module was supplied with the weighted values given by the Graph-based Keyword Extraction module as instead of the results of the Frequency Counting module. In this circumstance, tf and idf would not stand for term-frequency and inverse-document-frequency, but instead for term or inverse-document weighted scores. This condition can be regarded as a kind of integrated model of graph analysis and VSM.

The second model differed from the first in that the tf-idf weighting module received the frequencies for the keywords from the Frequency Counting module. This type of integration allowed the term-document matrix to be constructed in the way typical of tf-idf systems but the number of the rows was reduced because, same as in the first model, only a subset of the terms

were selected during the Keyword Extraction process.

The final experiment condition was a typical form of VSM and provided a control measure of semantic similarity by using the traditional method, as described in Section 2. This model did not implement the graph-ranking algorithm and skipped the Keyword Extraction stage.

### 3.3 Data

To collect the test data set, 30 famous objects on a list of Seoul cultural assets, each including some descriptive text, were selected for use as queries. For each query, 20 related documents were manually collected including Wikipedia documents, blog posts, and news articles written on the object. Two hundred texts unrelated to any of the queries were searched and stored. These texts consisted of articles, blogs, and web page texts on various topics but limited to social, economic or cultural contents. Hence, each of the cultural objects there would be 20 semantically related documents against 780 unrelated texts.

### 3.4 Evaluation Scheme

To estimate the performance of the models in this system, the well-known measure Mean Average Precision was used (Voorhees and Harman, 2005). This measure ranges from 0 to 1 where the maximum value 1 means that all target documents are placed higher than the non-related texts in the ordered output list.

## 4 Results

### 4.1 Morpheme vs. Word

For the 30 queries, given a word-window of 4 and a Proportion of Keywords of 0.4, the mean average of the precision (MAP score) for the morpheme-based criteria was 0.83 while the word-based criteria was 0.74.

Only four of the 30 queries showed higher MAP scores for the word-based separation method. Thus, in this study, the morpheme-based approach significantly outperforms the word-based approach.

### 4.2 Window Size

A Determining the length of word-window is related to the problem of morpheme/word based separation. In some languages, especially agglutinative languages like Korean, one word might be composed of more than three morphemes. Practically, this means that if there are two words both containing a noun and split by whitespace,

the probability of finding a morpheme between of them will be higher than in non-agglutinative languages such as English, widening the mean distance between any two nouns.

To confirm this prediction, an experiment manipulating the size of the window was conducted and the result is presented in Fig. 2. As one can see, the highest the MAP value (0.83) for the morpheme-based split method is obtained with a window length of 4. This pattern is what was expected given the discussion in Section 2.2. In contrast, using the word-based separation method, there does not seem to be any significant relationship between window size N and MAP score.

One easy interpretation of this result is that the word-based solution is not effective enough to capture the connective pattern of the terms in the network since it is missing the syntactic cues associated with words' stems.

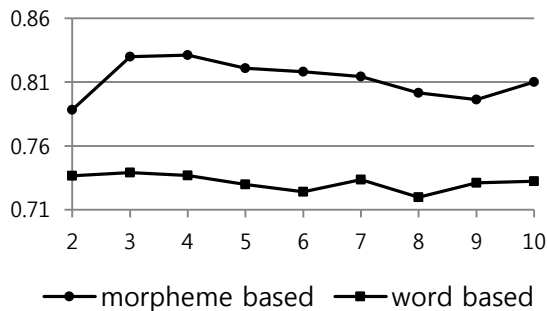


Figure 2. Mean average precision score in terms of window of word N.

### 4.3 Proportion of Keywords

In Fig. 3, the window size was set to 4 and the highest performing experiment model was used (the comparison result for the different models will be provided in next section).

As one can see in the graph, a proportion of 4/10 recorded the highest score. However, after this point the MAP score rapidly dropped; in contrast with the period of gradual increase observed up to that point.

To understand this result, it is important to recall that the tf-idf weighting mechanism uses inverse document frequency to give weights to the terms that are found in only a small number of documents but are frequent within a particular document. Hence, removing too many terms from the text would artificially increase the value of the idf component of tf-idf as a word may be rarely selected as a keyword despite occurring in a large number of documents.

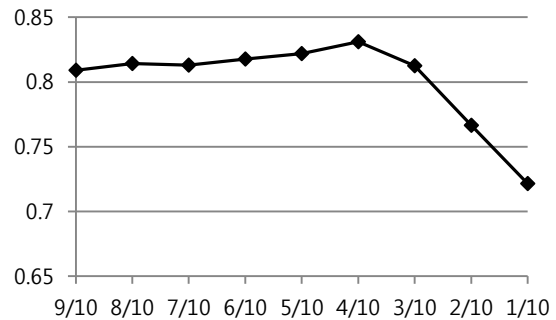


Figure 3. Mean average precision scores for the change of the keyword proportion (e.g., 9/10 means that 90 percent of the candidate nouns were accepted).

### 4.4 Comparison of Different Models

In Table 1, the notation [+/-] indicates whether the function was employed in the workflow of the experiment. Thus, Keyword extraction with plus sign means that the PageRank-based ranking algorithm was used for the keyword extraction process. Similarly, if the tf-idf module is displayed with minus sign then it means that frequencies per each term in the documents were replaced with the values from the graph-based analysis module.

Model	MAP score
Keyword extraction +, tf-idf -	0.81
Keyword extraction -, tf-idf +	0.80
Keyword extraction +, tf-idf +	<b>0.83</b>

Table 1. The MAP scores for different experiment conditions.

The results in Table 1 reveal that the full model (including all the sub-modules) outperforms the other two models, proving the research assumption that pre-filtering input texts would improve the quality of semantic similarity measurements based on VSM. The other modified condition employing the graph-analysis recorded the second highest, but the difference to the control condition was very small. These results were obtained using a window size of 4 and a keyword proportion of 4 out of 10; these values yielded the best outcome from the experiments.

In short, the comparative result of the experimental conditions suggest that drawing a bag of filtered words per document before tf-idf weighting could improve the process of computing semantic relatedness.



## Acknowledgments

The authors would like to thank the three anonymous reviewers for their helpful and valuable comments.

## References

- Brin, Sergey, & Page, Lawrence. (1998). The anatomy of a large-scale hypertextual Web search engine. Paper presented at the Proceedings of the seventh international conference on World Wide Web 7, Brisbane, Australia.
- Buckley, Chris. (2005). Project at TREC. In Ellen M. Voorhees & Donna K. Harman (Eds.), *TREC : experiment and evaluation in information retrieval* (pp. 301-320). Cambridge, Mass.: MIT Press.
- Erkan, Gunes, & Radev, Dragomir R. (2004). LexPageRank: Prestige In Multi-Document Text Summarization. Paper presented at the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.
- Grineva, Maria, Grinev, Maxim, & Lizorkin, Dmitry. (2009). Extracting key terms from noisy and multi-theme documents. Paper presented at the Proceedings of the 18th international conference on World wide web, Madrid, Spain.
- Kurland, Oren, & Lee, Lillian. (2006). Respect my authority!: HITS without hyperlinks, utilizing cluster-based language models. Paper presented at the Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA.
- Manning, Christopher D., Raghavan, Prabhakar, & Schtze, Hinrich. (2008). *Introduction to Information Retrieval*: Cambridge University Press.
- Mihalcea, Rada, & Tarau, Paul. (2004). TextRank: Bringing Order into Texts. Paper presented at the Conference on Empirical Methods in Natural Language Processing.
- Salton, Gerard M. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*: Prentice-Hall, Inc.
- Salton, Gerard M., & Buckley, Christopher. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5), 513-523. doi: 10.1016/0306-4573(88)90021-0
- Salton, Gerard M., Wong, Andrew, & Yang, ChungShu. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11), 613-620. doi: 10.1145/361219.361220
- Turney, Peter D. (2006). Similarity of Semantic Relations. *Comput. Linguist.*, 32(3), 379-416. doi: 10.1162/coli.2006.32.3.379
- Turney, Peter D., & Pantel, Patrick. (2010). From frequency to meaning: vector space models of semantics. *J. Artif. Int. Res.*, 37(1), 141-188.
- Voorhees, Ellen M., & Harman, Donna K. (2005). *Project at TREC*. Cambridge, Mass.: MIT Press.
- Wang, Jinghua, Liu, Jianyi, & Wang, Cong. (2007). Keyword extraction based on pagerank. *Advances in Knowledge Discovery and Data Mining*, 857-864.
- Widdows, Dominic, & Dorow, Beate. (2002). A graph model for unsupervised lexical acquisition. Paper presented at the Proceedings of the 19th international conference on Computational linguistics - Volume 1, Taipei, Taiwan.

# Semi-supervised Classification of Twitter Messages for Organization Name Disambiguation

Shu Zhang<sup>1</sup>, Jianwei Wu<sup>2</sup>, Dequan Zheng<sup>2</sup>, Yao Meng<sup>1</sup> and Hao Yu<sup>1</sup>

<sup>1</sup> Fujitsu Research and Development Center, Beijing, China

{zhangshu, mengyao, yu}@cn.fujitsu.com

<sup>2</sup> School of Computer Science and Technology, Harbin Institute of Technology,  
Harbin, China

{jwwu, dqzheng}@mtlab.hit.edu.cn

## Abstract

In this paper, we probe the problem of organization name disambiguation on twitter messages. This task is challenging due to the fact of lacking sufficient information in a tweet message. Instead of conventional methods based on mining external information from web sources to enrich information about organization, we propose to mine the relationship among tweets in data set to utilize context information for disambiguation. With a small scale of labeled tweets, we propose LP-based and TSVM-based semi-supervised methods to classify tweets. We aim to mine both related and non-related information for a given organization. The experiments on WePS-3 show that proposed methods are effective.

## 1 Introduction

Twitter is an online social networking and microblogging service, which rapidly gained worldwide popularity. How to retrieval, analyze and monitor Twitter information has been receiving a lot of attention in natural language processing and information retrieval research community (Kwak, *et al.*, 2010; Boyd, *et al.*, 2010; Tsagkias, *et al.*, 2011). One of the essential things of these researches is first to get the information which is related to the studied entity. This is caused by the ambiguity of entities. For example, the name of company “Apple” has a separate meaning referring to one kind of fruit. The word “Amazon” also could refer to river or company.

In this paper, we focus on finding related tweets for a given organization, which can be treated as a binary classification problem. Assuming that tweets are retrieved by a query, such as “apple”, the task is to classify whether each

retrieved tweet is relevant to the target organization (“Apple Inc.”) or not. However, constructing such a classifier is a challenging task, as tweets are short and informal. Additionally, the information about a given organization is limited, which is difficult to cover the word occurrences in the given organization related tweets.

Different from previous work on mining external information from web sources to enrich information about the given organization, we propose to mine the relationship among retrieved tweets in data set. With a small scale of labeled tweets, we propose semi-supervised methods to mine the relationships between labeled and unlabeled tweets for the given organization.

The remainder of the paper is organized as follows: Section 2 describes the related work on name disambiguation. Section 3 gives problem description and an overview of our approach. Section 4 and Section 5 present LP-based and TSVM-based semi-supervised methods to classify tweets. Section 6 gives the experiments and results. Finally section 7 summarizes this paper.

## 2 Related Work

Twitter contains little information in each tweet, with no more than 140 characters. This makes the tasks of analyzing Twitter messages more challenge, and attracts much interest from the research community in recent years (Meij *et al.*, 2012; Liu *et al.*, 2011; Sriram *et al.*, 2011).

The most related works are WePS-3 Online Reputation Management<sup>1</sup> held in 2010, which aims to identify tweets which are related to a given company (Amigó *et al.*, 2010).

In WePS-3, the research of (Yerva *et al.*, 2010) shows the best performance in the evaluation

---

<sup>1</sup> <http://nlp.uned.es/weps/>

campaign. They adopt SVM classifier with external resources, including Wordnet, metadata profile, category profile, Google set, and user feedback, to enrich the information of the given organization. Yoshida *et al.* (2010) classify organization names into “organization-like names” or “general-word-like names”. Kalmar (2010) adopts bootstrapping method to classify the tweets. The research of (García-Cumbreras *et al.*, 2010) shows the named entities in tweets are appropriate for certain company names.

There are some similar works. Perez-Tellez *et al.* (2011) adopt clustering technique to solve the problem of organization name disambiguation. Focus on identifying relevant tweets for social TV, Dan *et al.* (2011) propose a bootstrapping algorithm utilizing a small manually labeled dataset, and a large dataset of unlabeled messages.

Different from their works, we utilize semi-supervised methods to classify the tweets. We aim to transfer related or unrelated information of the given organization among tweets based on a small scale of labeled data.

Compared with bootstrapping algorithm, which is based on a local consistency assumption, LP algorithm is based on a global consistency assumption, and can effectively capture the natural clustering structure in both the labeled and unlabeled data to smooth the labeling function.

### 3 Overview

#### 3.1 Problem Statement

Given a set of tweets and an organization name, the goal is to decide whether each tweet in the set talks about the given organization or not.

In detail, the input information per tweet contains: the tweet identifier, the entity name, the query used to retrieve the tweet, the author identifier and the tweet content.

For each organization in the dataset, it gives the organization name and its homepage URL.

The output per tweet is True or False tag corresponding to related or non-related with the given organization.

#### 3.2 Our Method

In this paper, we propose semi-supervised methods to classify tweets for a given organization. This is considered from the following two points:

- Organization information automatically mined from web pages is limited, which could not cover the potentially infinite words occurred in tweets. However, how to

mine the high quality organization related information is also a problem.

- Both positive and negative samples are important for classification task. Though, it is possible to mine organization related information as positive sample from web by some key words or human input. However, it is difficult to obtain negative information about the other meanings of the given organization name which do not refer to the given organization.

Therefore, instead of mining external information from web sources to enrich information about organization, we propose to mine information directly from tweet set. The organization related information is extracted from the positive samples, which reflects keywords related to the given organization in tweets. The information extracted from the negative samples, gives the possible different interpretations of the given organization name.

With a small scale of labeled tweets for a given organization name, we utilize LP and TSVM based semi-supervised classifiers to mine unlabeled tweets, which will be described in the following section in detail.

### 4 LP Based Semi-supervised Classifier

Label Propagation (LP) is a graph-based semi-supervised algorithm, proposed by Zhu *et al.* (2002). The main idea of graph-based semi-supervised learning is to use pair-wise similarities between instances to enhance classification accuracy. It is a diffusion process on graphs, where the information is propagated from the labeled instances to the rest of unlabeled instances.

LP algorithm is to represent labeled (served as seeds) and unlabeled examples as nodes in a connected graph, then propagating the label information from any vertex to nearby nodes through weighted edges iteratively, finally get the labels of unlabeled examples after the propagation process converges. The labels of unlabeled examples are determined by considering both the similarity between labeled and unlabeled examples, and the similarity between unlabeled examples (Chen, *et al.*, 2008).

LP algorithm has achieved good performance in many applications, such as noun phrase anaphoricity in coreference resolution (Zhou, *et al.*, 2009), word sense disambiguation (Niu, *et al.*, 2005) and entity relation extraction (Chen, *et al.*, 2006).

## 4.1 Graph Building

Let  $X = \{x_i\}_{i=1}^n$  be a set of tweets for a given organization, where  $x_i$  represents  $i$ th tweet,  $n$  is the total number of tweets. The first  $l$  tweets are labeled  $(x_1, y_1) \dots (x_l, y_l)$ ,  $Y_L = \{y_1, \dots, y_l\}$  are labels. Here,  $y_i \in C$ ,  $C$  refers to two known classes (*True* or *False*) for this task. The others  $(x_{l+1}, y_{l+1}) \dots (x_{l+u}, y_{l+u})$  are the unlabeled tweets, where  $Y_U = \{y_{l+1}, \dots, y_{l+u}\}$  are unknown.

For the graph, the nodes represent both labeled and unlabeled tweets. The edge between any two nodes  $x_i$  and  $x_j$  is weighted by some distance measure. Based on assumption, the closer the two nodes are in some distance measure, the larger the weight  $w_{ij}$ , which is defined as follows:

$$w_{ij} = \exp\left(-\frac{s_{ij}^2}{\sigma^2}\right) = \exp\left(-\frac{\text{Cos}^2(x_i, x_j)}{\sigma^2}\right)$$

Where  $s_{ij}$  is the distance measure, we adopt cosine similarity to measure two nodes  $x_i$  and  $x_j$ .  $\sigma$  is a constant parameter to scale the weights.

For measuring the similarity of two nodes, we adopt two types of features to represent each tweet: one is the unigram word unit, the other is 4-gram character unit.

Unigram word: the words contain in a tweet after filtering stop words.

4-gram character unit: the possible 4-gram character for each unigram word.

The tweet is short and informal. There are little information contain in one tweet. One key-word missing may lead the change of the tweet's classification result. Therefore, we adopt character unit as feature to allow the mistake of spelling in some extent.

## 4.2 Algorithm

All nodes in graph have soft labels that can be interpreted as distribution over labels. The label of a node is propagated to all nodes through the edges. Larger edge weights allow labels to travel through easier. Define a  $n \times n$  probabilistic transition matrix  $T$ , ( $n = l + u$ ).

$$T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}}$$

Here  $T_{ij}$  is the probability to jump from node  $j$  to node  $i$ . We define a  $(l+u) \times C$  label matrix  $Y$ , the  $i$ th row representing the label probability distribution of node  $x_i$ .

The label propagation algorithm is as follows:

- (1) Propagate  $Y \leftarrow TY$
- (2) Row-normalize  $Y$ , to maintain the label probability interpretation
- (3) Clamp the labeled tweets, replace the  $Y_L$  with the initial value
- (4) Repeat from step (1) to (3) until  $Y$  converges

Here, we make use of JUNTO Label Propagation toolkit<sup>2</sup> to implement this algorithm.

## 5 Transductive SVM

Transductive Support Vector Machines (TSVM) is a semi-supervised learning method, which can be treated as an extension of SVM by introducing unlabeled data. Similar with SVM, TSVM tries to label the unlabeled data, and find the maximum margin separation hypersurface that separates the positive and negative instances of labeled data and the unlabeled data. The basic idea of TSVM is to seek a decision surface away from the dense regions of unlabeled data.

For the given labeled tweets  $\{(x_i, y_i) | x_i \in R^n, y_i \in \{-1, +1\}\}_{i=1}^L$ ,  $y_i$  refers to two known classes (*True* or *False*) for this task, and unlabeled data  $\{x_j^* | x_j^* \in R^n\}_{j=1}^{L^*}$ . This can be written as minimizing

$$\arg \min_{w, b, \xi, y^*} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^L \xi_i + C^* \sum_{j=1}^{L^*} \xi_j^*$$

Subject to

$$\forall i: y_i w \bullet \phi(x_i) + b \geq 1 - \xi_i$$

$$\forall j: y_j^* w \bullet \phi(x_j) + b \geq 1 - \xi_j^*$$

$$\forall i: \xi_i \geq 0$$

$$\forall j: \xi_j^* \geq 0$$

$$\forall j: y_j^* \in \{-1, +1\}$$

$$w \in R^m, b \in R$$

Similar with LP, we adopt two types of features to represent each tweet: one is the unigram word unit, the other is 4-gram character unit. Here, we make use of SVMLight tools to implement this algorithm.

## 6 Experiments and Results

### 6.1 Corpus and Evaluation Metric

We have conducted experiments on the WePS-3 task 2 data. The test data contain about 50 organization names with about 450 tweets for each organization.

<sup>2</sup> <http://code.google.com/p/junto/>

	<i>P+</i>	<i>R+</i>	<i>F+</i>	<i>P-</i>	<i>R-</i>	<i>F-</i>
LP	0.8097	0.5008	0.4120	0.8059	0.5593	0.5166
TSVM	0.6683	0.6969	<b>0.7144</b>	0.6942	0.6484	<b>0.6972</b>
Top_1	0.7108	0.7445	0.6264	0.8443	0.5195	0.5606
Top_2	0.7546	0.5409	0.4935	0.7413	0.6049	0.5651
Top_3	0.7410	0.6157	0.5062	0.7365	0.4911	0.4683
Baseline (NR)	1.0000	0.0000	0.0000	0.5652	1.0000	0.6563
Baseline (R)	0.4348	1.0000	0.5274	1.0000	0.0000	0.0000

Table 1. Performances of semi-supervised methods and other systems

The task is to classify the tweets related or non-related with the given organization, it belongs to classification task. Therefore, we measure the performance by *accuracy*, *precision*, *recall* and *F-measure*.

## 6.2 Results and Analysis

Based on the test data, we testify the performance of our proposed methods.

### Seed selection for semi-supervised classifiers

We random select 100 tweets as seeds from the test data for each organization name, which is about 20% for tweet set.

Decreasing the influence of seed selection for the performances of semi-supervised classifiers, we try out the experiments five times and get the average values for the final results.

### Performance of semi-supervised classifiers

For comparison, we select five system results as references, three of them are the top 3 systems in WePS contest, the other two systems are the baseline systems. Two baseline systems tag all tweets as related (Baseline (R)) or non-related (Baseline (NR)) to each organization.

Figure 1 and Table 1 show the performances of semi-supervised methods and other systems.

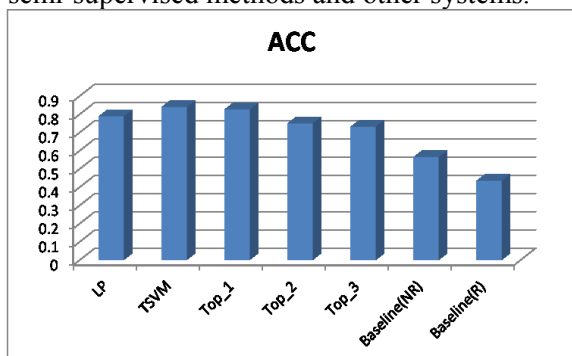


Figure 1. Accuracies of semi-supervised methods and other systems

In Figure 1, the *accuracy* of Baseline (NR) is higher than that of Baseline (R), which shows

there are more unrelated tweets in the whole test data, the disambiguation of tweets is necessary. The accuracies of our proposed methods and Top 3 systems are all much higher than those of two baselines. It proves that adopting some methods to disambiguate tweets is feasible.

The accuracies of our proposed semi-supervised methods based on LP and TSVM are both higher than that of Top\_2 system. The accuracy of TSVM is 0.8391, which is higher than that of Top\_1 (0.8267). It proves that semi-supervised methods are effective for this task. Instead of mining web sources, it is also effective to mine the information among tweets, especially which including both related and non-related information about the organization name.

In Table 1, it shows the performance of each system on *precision*, *recall* and *F-measure*. The values are calculated on the average performance for each organization name in test data set. Though *P+* and *R+* values of TSVM are not the highest ones, the *F+* value is highest among the five systems. *F-* value is also the highest one. it shows that TSVM-based classifier gets the best balance between precision and recall for classification. *F+* value is important to measure the ability of finding the related tweets to a given organization.

## 7 Conclusion

In this paper, we probe the problem of organization name disambiguation on twitter information. We utilize LP and TSVM based semi-supervised method to implement the disambiguation system. The experiments on WePS-3 show that both LP-based classifier and TSVM-based classifier are effective. Especially, TSVM-based classifier gets higher performance than that of the best result in WePS contest, which proves that semi-supervised method is a feasible way to classify the related tweets information for a given organization on Twitter.

## References

- Enrique Amigó, Javier Artiles, Julio Gonzalo, Damiano Spina, Bing Liu, and Adolfo Corujo. 2010. WePS-3 Evaluation Campaign: Overview of the Online Reputation Management Task. In Proceedings of 3rd Web People Search Evaluation Workshop.
- Surender R. Yerva, Zoltán Miklós, and Karl Aberer. 2010. It was Easy, when Apples and Blackberries were only Fruits. In Proceedings of 3rd Web People Search Evaluation Workshop.
- Minoru Yoshida, Shin Matsushima, Shingo Ono, Issei Sato, and Hiroshi Nakagawa. 2010. ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In Proceedings of 3rd Web People Search Evaluation Workshop.
- Paul Kalmar. 2010. Bootstrapping Websites for Classification of Organization Names on Twitter. In Proceedings of 3rd Web People Search Evaluation Workshop.
- M.A. García-Cumbreras, M. García-Vega M, F. Martínez-Santiago and J.M. Peréa-Ortega. 2010. SINAI at WePS-3: Online Reputation Management. In Proceedings of 3rd Web People Search Evaluation Workshop.
- Fernando Perez-Tellez, David Pinto, John Cardiff, and Paolo Rosso. 2011. On the Difficulty of Clustering Microblog Texts for Online Reputation Management. In Proceedings of 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT.
- Ovidiu Dan, Junlan Feng, and Brian D. Davison. 2011. A Bootstrapping Approach to Identifying Relevant Tweets for Social TV. In Proceedings of 5th International AAAI Conference Weblogs and Social Media.
- 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 Proceeding of 17th WWW, pages 91-100.
- Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a Social Network or a News Media? In Proceeding of 19th WWW, pages 591-600.
- Danah Boyd, Scott Golder, and Gilad Lotan. 2010. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In Hawaii International Conference on System Sciences, pages 1-10.
- Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. 2011. Linking Online News and Social Media. In Proceedings of 4th ACM Web Search and Data Mining, pages 565-574.
- Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. 2012. Adding Semantic to Microblog Posts. In proceedings of 5th ACM Web Search and Data Mining, pages 563-572.
- Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing Named Entities in Tweets. In Proceedings of 49th Annual Meeting of the Association for Computational Linguistics, pages 359-367.
- Bharath Sriram, David Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2011. Short Text Classification in Twitter to Improve Information Filtering. In Proceedings of the ACM SIGIR 2011, pages 841-842.
- Guo Dong Zhou and Fang Kong. 2009. Global Learning of Noun Phrase Anaphoricity in Coreference Resolution via Label Propagation. In Proceedings of Empirical Methods in Natural Language Processing, pages 978-986.
- Zheng Yu Niu, Dong Hong Ji, and Chew Lim Tan. 2005. Word Sense Disambiguation Using Label Propagation Based Semi-Supervised Learning. In Proceedings of 43rd Annual Meeting on Association for Computational Linguistics, pages 395-402.
- Jin Xiu Chen, Dong Hong Ji, Chew Lim Tan, and Zheng Yu Niu. 2006. Relation Extraction Using Label Propagation Based Semi-supervised Learning. In Proceedings of 21st International Conference on Computational Linguistics and 44th Annual Meeting on Association for Computational Linguistics, pages 129-136.
- Jin Xiu Chen, and Dong Hong Ji. 2008. Graph-Based Semi-supervised Relation Extraction. In Journal of Software, 19(11): 2843-2852.
- Xiao Jin Zhu and Zou Bin Ghahramani. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University

# Word in a Dictionary is used by Numerous Users

**Eiji Aramaki**

Kyoto University/PRESTO  
[eiji.aramaki@gmail.com](mailto:eiji.aramaki@gmail.com)

**Sachiko Maskawa**

Photonic System Solutions  
[sachiko.maskawa@gmail.com](mailto:sachiko.maskawa@gmail.com)

**Mai Miyabe**

Kyoto University  
[mai.miyabe@gmail.com](mailto:mai.miyabe@gmail.com)

**Mizuki Morita**

University of Tokyo  
[morita.mizuki@gmail.com](mailto:morita.mizuki@gmail.com)

**Sachi Yasuda**

NINJAL  
[yasudasac@gmail.com](mailto:yasudasac@gmail.com)

## Abstract

Dictionary editing requires enormous time to discuss whether a word should be listed in a dictionary or not. So as to define a dictionary word, this study employs the number of word users as a novel metrics for selecting a dictionary word. In order to obtain the word user, we used about 0.25 billion tweets of approximately 100,000 people published for five months. This study compared the classification performance of various measures. The result of the experiments revealed that a word in a dictionary is used by numerous users.

## 1 Introduction

Dictionary editing requires numerous time to discuss whether a word should be listed in a dictionary or not. In order to define a dictionary word, this study assumes that the following two scales are essential than word frequency:

- (1) **Usage period:** a dictionary word has been used for larger period of time.
- (2) **User population:** a dictionary word has been used by more people.

The first scale is hard to measure in practice, because the usage period requires a longitudinal data. For the investigation, the second clue has more feasibility by social media resources, which enables to know the word usage for each user.

The objective of this study is to retrieve a dictionary word. This study approaches this problem by drawing on a binary classification task, which divides the words into two categories:

ries: a dictionary word (listed in a dictionary) and a out-of-dictionary word.

For the database, we have collected 0.25 billion tweets of 100,000 people from Twitter. The experimental results have revealed that a dictionary word is highly correlated with the number of word users. Although the experiment is conducted in Japanese language, the proposed method does not depend on a specified language.

## 2 Related Work

So far, a strong clue for dictionary editing is a word frequency. The relation between a word frequency and its coverage has been an interest for many researchers (Crowley 2003, Freeborn 2006, Burrige and Kortmann 2008). In English, the frequent 2,000 words cover 90% of spoken language (West 1953), and the most frequent 6,000 words cover 90% of written language (Francis, Kučera et al. 1982). The results in Japanese are similar to them. The frequent 10,000 words cover almost all vocabulary used in magazines (90 magazines) (NINJAL 1997), and 17,000 words cover the vocabulary spoken in television programs (Ishino 2000). Although the target media differs, they share the same findings that frequent words cover most of the corpus. This study presents another word measure.

## 3 Materials

This study has used two types of data: user corpus (Section 3.1), and a gold standard data (Section 3.2):

### 3.1 Corpus: 100,000 people tweets

This study employs Twitter as a fundamental database, because Twitter has two strong advantages for the purpose of this study: (1) it has numerous

users and (2) the author information is available for each tweet. This study sampled 0.25 billion tweets from 99,964 people, as described below.

- **Data collection period:** 143 days from November 3<sup>rd</sup> 2009 to March 25<sup>th</sup> 2010.
- **Number of users:** 99,964 people, as extracted based on the following three qualifications:
  - A user who posts at least 5 tweets per a month
  - Total posts contain over 5,000 words.
  - Japanese language users: at least one Japanese UTF code characters are used in the first tweet.
- **Total number of tweets:** 253,482,784 tweets (4,258,707,255 words): the words are analyzed by a morphological analyzer (Kurohashi, Nakamura et al. 1994)

### 3.2 Gold standard Data

The gold standard data of this study is a word listed in the *IWANAMI* Japanese Dictionary 7<sup>th</sup> edition (Nishio, Iwabuchi et al. 2009). This dictionary is one of the best selling dictionaries in Japanese.

## 4 Methods

The task of this study is to classify whether a word is listed in a dictionary or not. For the classification, this study employs four measures:

1.  $freq(w)$ : a word frequency of a word  $w$ .
2.  $R_{freq}(w)$ : a rank of  $freq(w)$ .
3.  $user(w)$ : the number of users of a word  $w$ .
4.  $R_{user}(w)$ : a rank of  $user(w)$ .

While the first two ( $freq(w)$  and  $R_{freq}(w)$ ) are conventional measures used among the many previous researches, the other two ( $user(w)$  and  $R_{user}(w)$ ) are newly introduced by this study.

### Baseline Approach

A easy approach is to select a word which has enough frequency (more than  $\alpha$  times). This approach is formalized as follows:  $freq(w) > \alpha$ .

### Proposed Approach

Instead of the frequency, the proposed approach relied on the number of users ( $user(w)$ ). This approach is formalized as follows:  $user(w) > \alpha$

### Another Proposed Approach

This approach makes balance between the number of users ( $user(w)$ ) and the word frequency ( $freq(w)$ ). If both measures stay in balance, the both ranks should equal, satisfying the following formula:

$$R_{user}(w) = R_{freq}(w) .$$

If a certain user prefers to use specific words, the rank of the frequency ( $R_{freq}$ ) become larger than that of users ( $R_{users}$ ):

$$R_{user}(w) > R_{freq}(w) .$$

In the same method, a widely used word could be extracted by using the following formula:

$$R_{user}(w) < R_{freq}(w) .$$

## 5 Experiment

### 5.1 Test-set: Wikipedia entry names

A test-set consists of 4,000 nouns, which are randomly sampled from Wikipedia entry names. Half of them (2,598 nouns) are listed in the dictionary (positive examples). The other 1,402 words are out-of-dictionary (negative examples).

### 5.2 Comparable Methods

We compared the following classification methods:

- **Rfreq:** this method selects the words whose frequency is in the top  $\alpha$  rank:  $R_{freq}(w) < \alpha$ .
- **Ruser:** this method selects the words whose user size is in the top  $\alpha$  rank:  $R_{user}(w) < \alpha$ .
- **Ruser' (weighted based):** this method is essentially based on the number of users. However, it is weighted by the frequency as follows:  $-\log(freq(w)) \cdot user(w) < \alpha$ .
- **Ruser/Rfreq:** this approach is based on the balance of two ranks:  $R-Ratio < \alpha$ . Here,  $R-Ratio = R_{user}(w) / R_{freq}(w)$ .

The evaluation is conducted in possible  $\alpha$  range ( $\alpha = 0 \sim \infty$ ).

### 5.3 Evaluation Metrics

The methods are evaluated using information retrieval metrics:

- **Precision (P):** # of correct outputs / # of system positive outputs.
- **Recall (R):** # of correct outputs / # of positive examples (=2,598).
- **F-measure (F):** harmonic mean between the precision and the recall.



## 5.4 Result

The precision-recall curve for each method is presented in Figure 1. The best F-measure points of all methods are the same (Recall=1; Precision=0.6). However, the accuracies differ in the low-recall area. Basically **Ruser** (partly **Ruser/Rfreq**) showed the best performance. **Rfreq** constantly showed poor performance rather than the others. These results indicated that the number of users is an essential factor.

Figure 2 shows the distribution of dictionary words plots in  $R_{freq}(w)$  and  $R_{user}(w)$ . Numerous words are distributed on the balanced line ( $X=Y$ ), indicating that  $R_{freq}(w)$  and  $R_{user}(w)$  correlated with each other.

We found several outliers in the TOP-LEFT area ( $Y \gg X$ ), suggesting that several words have the low number of users compared to the frequency metrics. The examples of such words are presented in Table 1 (b), consisting of many out-of-dictionary words.

## 5.5 Discussion

This study reveals that the number of users is an important clue to classify a dictionary word. This result has a number of applications; e.g., the popular vocabulary learning, a user number-based spell checking system, and so on.

However, this study has several limitations, which comes from the following factors:

- **User bias:** Most Twitter users are 20-30 years old. This population gap might bias the results.
- **Device bias:** The type of input device, such as keyboard typing, touch pad, and input suggestion, might bias the results.
- **Twitter bias:** The length limit of Twitter (140 characters) might prefers shorter words.

Reducing the above biases is one of the remaining problems.

## 6 Conclusion

This study proposes a method to classify a dictionary word. We assume that a dictionary word should be used by many users. To prove this point, we have obtained the 100,000 user texts from Twitter. Then, we have evaluated various measures: a frequency based, a user based, and the ratio based. The experimental result has revealed that the number of word users is an essential indicator for classifying a dictionary word.

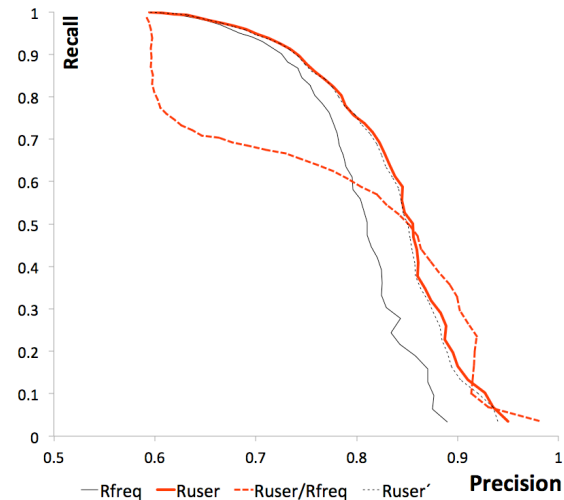


Figure 1: The precision-recall curve for each method.

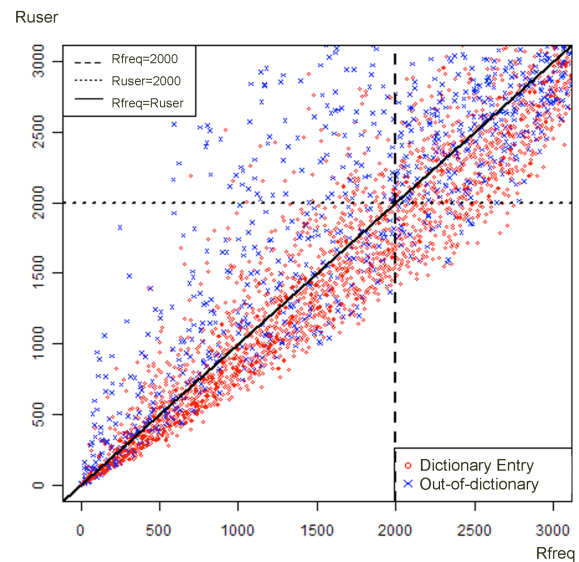


Figure 2: The Rank of Word User ( $R_{user}$ ) and the Rank of Word Frequency ( $R_{freq}$ ).

The X-axis indicates the rank of word frequency ( $R_{freq}$ ); the Y-axis indicates the rank of word users. The dotted line indicates  $R_{freq}=2000$  and  $R_{user}=2000$ . The line indicates that the balanced line ( $R_{freq}=R_{user}$ ). The RIGHT-BOTTOM area contains words that are high user words. The LEFT-TOP area contains words that are low user words. As shown in the figure, most of low user words are out-of-dictionary words.

Smith, J. (1996). An Historical Study of English: Function, Form and Change. London, Routledge.  
 West, M. (1953). A General Service List of English Words, Longman.

Table 1: Word Example of  $R_{user}/R_{freq}$ .

(a) High, and (b) Low

(a) Low R-Ratio

w	freq(w)	Rfreq(w)	user(w)	Ruser(w)	R-RATIO
週間 <i>week</i>	379183	554	77943	282	0.5
復活 <i>restore</i>	293265	697	70103	392	0.56
予定 <i>plan</i>	917124	243	88721	146	0.6
気分 <i>mood</i>	601588	351	82794	211	0.6
昨日 <i>yesterday</i>	1519917	160	93673	97	0.6
原因 <i>reason</i>	212165	958	60124	619	0.64
決定 <i>decision</i>	320819	642	68865	417	0.64
時間 <i>time</i>	3933947	69	97927	45	0.65

(b) High R-Ratio

w	freq(w)	Rfreq(w)	user(w)	Ruser(w)	R-RATIO
旦那 <i>buddy</i>	210886	966	27914	2157	2.23
てら <i>hella</i>	315380	656	36352	1562	2.38
爆発 <i>burst-out</i>	581952	359	51831	867	2.41
原稿 <i>draft</i>	328386	634	34422	1680	2.64
たん *	792173	270	55067	747	2.76
ボク *	256087	792	24774	2396	3.02
おつ *	485454	431	39277	1398	3.24
ノシ *	352862	592	22786	2559	4.32

\* indicates a Japanese slang, which is hardly to translate.

Table 2: Low R-ratio words (out-of-dictionary).

w	freq(w)	Rfreq(w)	user(w)	Ruser(w)	R-RATIO
ダウンロード <i>download</i>	130200	1517	40373	1329	0.87
再起動 <i>reboot</i>	97634	1926	33090	1779	0.92
スイーツ <i>sweets</i>	74842	2420	26448	2272	0.93
マック <i>Mac</i>	231020	882	52983	821	0.93
ディズニー <i>Disney</i>	42470	3072	17394	2920	0.95
プレゼン <i>presentataion</i>	73507	2451	24264	2433	0.99
インストール <i>install</i>	147180	1360	39767	1365	1
アカウント <i>acount</i>	296177	690	55737	733	1.06
D S <i>D S</i>	193599	1033	43858	1167	1.12

## References

- Burridge, K. and B. Kortmann (2008). Varieties of English: vol 3, Berlin and NY: Mouton de Gruyter.  
 Crowley, T. (2003). Standard English and the Politics of Language, Palgrave Macmillan.  
 Francis, W. N., H. Kučera and A. W. Mackie (1982). Frequency analysis of English usage: lexicon and grammar, Houghton Mifflin.  
 Freeborn, D. (2006). From Old English to Standard English: A Course Book in Language Variations Across Time, Palgrave Macmillan.  
 Ishino, H. (2000). "studies in the Japanese language " Kokugogaku **51**(3): 41-47.  
 Kurohashi, S., T. Nakamura, Y. Matsumoto and M. Nagao (1994). Improvements of Japanese Morphological Analyzer JUMAN. The International Workshop on Sharable Natural Language Resources.  
 NINJAL (1997). The total vocabulary and their written forms in ninety magazines of today.  
 Nishio, M., E. Iwabuchi and S. Mizutani (2009). IWANAMI Japanese dictionary 7th, Iwanamishoten.

# Extracting Evaluative Conditions from Online Reviews: Toward Enhancing Opinion Mining

Yuki Nakayama

Atsushi Fujii

Department of Computer Science  
Graduate School of Information Science and Engineering  
Tokyo Institute of Technology  
{nakayama.y.aj@m, fujii@cs}.titech.ac.jp

## Abstract

A fundamental issue in opinion mining is to search a corpus for opinion units, which typically comprise the evaluation by an author for a target object from an aspect, such as “This hotel is in a good location”. However, no attempt has been made to address cases where the validity of an evaluation is restricted on a condition in the source text, such as “for traveling with small kids”. In this paper, we propose a method to extract such conditions, namely evaluative conditions, from sentences including opinion units. Our method uses supervised machine learning to determine whether each phrase is a constituent of an evaluative condition. We propose several features associated with lexical and syntactic information, and show their effectiveness experimentally.

## 1 Introduction

Reflecting the rapid growth in the use of opinionated texts on the Web, such as customer reviews, opinion mining has been explored to facilitate utilizing opinions mainly for improving products and decision-making purposes. While in a broad sense opinion mining refers to a process to discover useful knowledge latent in a corpus of opinionated texts, in a narrow sense its purpose is to extract opinions from a corpus. In either case, fundamental issues involve modeling a unit of opinions and searching the corpus for those units, which typically comprise the evaluation by an author for a target object from an aspect.

We take the following review sentence as an example opinionated description.

“I think hotel A is in a good location for traveling with small kids”.

From the above example, existing methods (Pang and Lee, 2008; Seki et al., 2009; Jin et al., 2009; Zhao et al., 2010; He et al., 2011; Liu and Zhang, 2012) for opinion mining extract the following quintuple as an opinion unit.

Target = “hotel A”, Aspect = “location”,  
Evaluation (Polarity) = “good” (positive), Holder = “I (author)”, Time = N/A

Depending on the application, “Evaluation” can be any of a literal evaluation expression (e.g., “good”), a polarity (positive/negative), or a value for multipoint scale rating. However, because this difference is not important in our research, we usually use the term “evaluation”.

Given those structured items extracted from a corpus, it is easy to overview the distribution of values for each element or a combination of elements. Those who intend to improve the quality of hotel A may investigate the distribution of values for “Aspect” in the reviews with “Target=hotel A & Polarity=negative”, while those who look for accommodation may compare the distribution of values for “Aspect & Polarity” in reviews for more than one hotel.

However, no attempt has been made to address cases where the validity of an evaluation is restricted on a condition in the source text. We shall call such a condition “evaluative condition”. In the above example sentence, the evaluation for hotel A (“in a good location”) is valid only “for traveling with small kids”, and it is not clear whether this evaluation is valid irrespective of the situation. The existing methods, which do not analyze evaluative conditions, potentially overestimate or underestimate the utility of hotel A and the quality of opinion mining is decreased accordingly.

To alleviate this problem, we need to introduce evaluative conditions as an element in the opinion unit, such as Condition=“for traveling with small kids”, which enables us to perform deeper

and finer-grained analysis for opinion mining. To avoid any confusion, we consistently use the term “opinion unit” to refer to the traditional quintuple-based unit in which a few elements can be omitted.

Motivated by the above background, in this paper we propose a method to extract evaluative conditions from opinionated corpora. The contribution of our research is introducing the notion of evaluative conditions into opinion mining and proposing a method to extract evaluative conditions from opinionated corpora.

Currently, we target corpora of review text in Japanese. As the first step of research, we focus only on cases where an evaluative condition and an opinion unit are in the same sentence. In addition, we leave the following two research issues as future work.

First, compared with the existing opinion elements, such as Aspect, values for Condition tend to be long and thus it is important to standardize various expressions for the same condition, such as “for traveling with small kids” and “for a family trip with children”. It can be expected that existing methods for paraphrasing alleviate this problem.

Second, it can be useful to subdivide evaluative conditions into general or domain-specific categories, such as “purpose”, “user”, and “situation” in reviews for hotels. For example, those categories can be effective to refine user’s needs in retrieving or recommending products. We show example sentences for several categories, in which the evaluative condition and evaluation expression are in bold and italic faces, respectively.

The room is *large enough* **for a business trip**. (purpose)

The bed is *small* **for people who is 185cm tall**. (user)

**If you stay more than one day**, you will be *tired of* the breakfast. (situation)

I was *content* with the meal **if it was less expensive**. (counterfactual)

**Considering the class of this hotel**, the dinner is *acceptable*. (concession)

## 2 Related Work

Evaluative conditions are related to causes and reasons because all of them have an influence on the validity of the corresponding evaluation in an opinion.

Although causal relations can be divided into inter-sentential and intra-sentential, our current interest is more related to the extraction of intra-sentential relations (Girju, 2003; Chang and Choi, 2004; Inui et al., 2005). These methods generally identify two event-related components in a sentence and determine the type of the causal relationship between those components, if any, such as “precondition”, “cause-effect”, and “consequence”. An event-related component is usually a word, such as “cancer”, or a proposition, such as “he is a heavy smoker”.

However, the above existing methods focused only on specific syntactic patterns, such as “<Clause1, Marker (*tame* in Japanese), Clause2>” (Inui et al., 2005) and “<NP1, Verb, NP2>” (Girju, 2003; Chang and Choi, 2004). In Section 1, none of the example sentences including evaluative conditions matches to those patterns, irrespective of whether in English or in Japanese. Additionally, looking at the examples for “counterfactual” and “concession”, the relation between the evaluation and evaluative condition is different from the causal relation. Besides this, our research is the first attempt to extract cause-like relations in opinion mining.

Kim and Hovy (2006) proposed a method to identify a reason for the evaluation in an opinion, such as “the service was terrible because the staff was rude” and “in a good location close to the station”. However, their purpose is to identify grounds that justify the evaluation, which are different from evaluative conditions.

## 3 Proposed method

### 3.1 Overview

The purpose of our method is to extract one or more evaluative conditions in an opinionated sentence in Japanese. Currently, we assume that both an opinion unit and an evaluative condition are in the input sentence, and that the opinion unit has been identified by an existing automatic method.

Our extraction method follows the BIO chunking classifier, which labels each token in a sentence as being the beginning (B), inside (I), or outside (O) of a span of interest. However, because there is no specific characteristics at the beginning of evaluative conditions in Japanese, we do not use the “B” label. We regard Japanese bunsetsu phrases, which consists of a content word and one or more postpositional particles, as tokens, and ex-

tract a sequence of I-phrases as an evaluative condition. However, phrases in an opinion unit are always classified into O-phrases. We use Support Vector Machine (SVM) to train a binary classifier for bunsetsu phrases and propose several features associated with lexical and syntactic information.

### 3.2 Features for phrase classification

Figure 1 depicts an example of syntactic dependency analysis for a review sentence in Japanese. We used “CaboCha” (Kudo and Matsumoto, 2002) for the dependency analysis. In Figure 1, a rectangle and an arrow denote a phrase and a dependency between two phrases, respectively, and in each phrase we show Romanized Japanese words and their English translations in parentheses.

Looking at Figure 1, by definition the evaluative condition (phrases #3-6) modifies the evaluation expression (phrase #7), but does not modify other opinion elements including the aspect (phrase #2). Also, the evaluative condition ends with specific particles in phrase #6. These properties motivated us to propose the following five features for the binary phrase classification.

**Feature A:** Because an evaluative condition modifies the evaluation expression, they are usually in close proximity to each other. Thus, there should be a pass of dependencies between an I-phrase and the evaluation expression, and a phrase in closer proximity to the evaluation expression is more likely to be an I-phrase. We use the dependency distance (i.e., the number of dependencies) between a phrase in question and the evaluation expression as the value for feature A. The value for a phrase is -1 if there is no pass between that phrase and the evaluation expression. In Figure 1, values for phrases #1, #4, and #8 are 2, 3, and -1, respectively.

**Feature B:** Feature A is not robust against errors of the dependency analysis. To complement this weakness of feature A, we roughly estimate the dependency distance by a phrase distance. In practice, we use the difference between the phrase IDs between a phrase in question and the evaluation expression as the value for feature B. If the evaluation expression consists of more than one phrase, we take the minimum difference. Because Japanese sentence has a post modification structure, in which a modifier is followed by its head, a phrase with a negative value for feature B is usu-

ally an O-phrase. In Figure 1, unlike the case for feature A, the values for phrase #1 is 6.

**Feature C:** Because an evaluative condition does not modify any opinion elements other than the evaluation expression, for the value of feature C we take 0 if there is a pass of dependencies between a phrase in question and a non-evaluation opinion element; otherwise 1. In Figure 1, values for phrases #1, #4, and #8 are 0, 1, and 1, respectively.

**Feature D:** Because an evaluative condition often ends with one or more specific particles, we use the existence (1/0) of those particles in a phrase as the value for feature D. Example particles include “ga (the nominative case)”, “no (of)”, “nitottte (for)”, and “nara (if)”. In Figure 1, values for phrases #1 and #6 are 1 and those for the remaining phrases are 0.

**Feature E:** As in Figure 1, an evaluative condition often consists of a phrase whose value for feature D is 1 and one or more preceding phrases. We use the existence (1/0) of a pass of dependencies between a phrase in question and a phrase whose value for feature D is 1. In Figure 1, values for phrases #3-5 are 1 and those for the remaining phrases are 0.

## 4 Experiments

To evaluate the effectiveness of our method, we used the Rakuten Travel data<sup>1</sup>, which consists of approximately 348,564 reviews for hotels in Japanese. From this data set, we randomly selected 675 reviews and manually annotated quintuples for opinion units and evaluative conditions. We found that 182 reviews include evaluative conditions and decomposed those reviews into sentences. We collected 286 sentences including evaluative conditions and used those sentences as the corpus for experiments. The total number of bunsetsu phrases in our corpus is 2,472, which consists of 761 I-phrases and 1,126 O-phrases in which 585 phrases are elements in opinion units.

We performed 10-fold cross-validation and compared different methods in terms of precision (P), recall (R), and F-measure (F). In Table 1, while “Phrase” denotes the result of the binary classification for bunsetsu phrases, “Condition” denotes that of extracting evaluative conditions as a whole using the BIO classifier. The line “Rule” denotes the result of a rule-based method, which is

<sup>1</sup><http://www.nii.ac.jp/cscenter/idr/rakuten/rakuten.html>

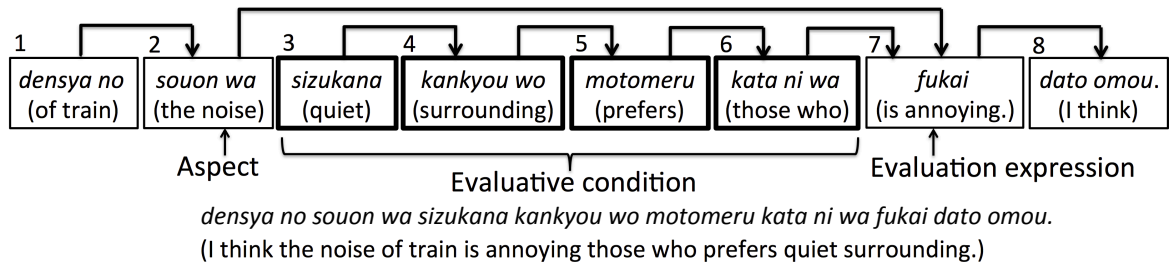


Figure 1: Example of dependency analysis for Japanese.

used as the baseline. This method extracts a bunsetsu phrase ending with one or more specific particles and all phrases from which there is a dependency path to that phrase. For example, in Figure 1 because phrase #6 ends with specific particles, the rule-based method extracts a sequence of phrases #3-#6 as an evaluative condition. The remaining lines denote different combinations of our five features in Section 3.2. While “w/o X” denotes our method without feature X, “All” denotes our complete methods using the five features.

Looking at Table 1, one can see that any variation of our method outperformed the rule-based method irrespective of the configuration, and that our complete method outperformed the remaining of our methods in terms of F-measure. We used the two-tailed paired t-test for statistical testing and found that the differences of “Rule” and “All” in F-measure for “Phrase” and “Condition” were significant at the 1% level. Thus, we conclude that each of our five features was independently effective for extracting evaluative conditions in review sentences and that when used together the improvement was even greater. At the same time, because values for P, R, and F in “Condition” were substantially smaller than those in “Phrase”, we need to improve methods to combine I-phrases and determine the final evaluative condition.

	Phrase			Condition		
	P	R	F	P	R	F
Rule	.539	.614	.553	.407	.412	.410
w/o A	.733	.797	.734	.505	.541	.517
w/o B	.598	.685	.609	.410	.460	.426
w/o C	.719	.789	.725	.490	.524	.500
w/o D	.732	.787	.732	.522	.554	.531
w/o E	.745	.756	.713	.456	.496	.468
All	.730	.792	.735	.538	.571	.548

Table 1: Results for experiments.

## 5 Conclusion

Although a number of methods have been proposed to search an opinionated corpus for opinion units, no attempt has been made to address cases where the validity of an evaluation in an opinion is restricted on a condition in the source text. We proposed a method to extract such conditions, namely evaluative conditions, from sentences including opinion units. Our method performed supervised binary classification to determine whether each phrase is a constituent of an evaluative condition. We proposed five features associated with lexical and syntactic information for Japanese, and show their effectiveness using reviews for hotels. Future work includes addressing research issues discussed in Section 1.

## References

- Du-Seong Chang and Key-Sun Choi. 2004. Causal relation extraction using cue phrase and lexical pair probabilities. In *Proceedings of 1st International Joint Conference on Natural Language Processing*, pages 61–70.
- Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83.
- Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 123–131.
- Takashi Inui, Kentaro Inui, and Yuji Matsumoto. 2005. Acquiring causal knowledge from text using the connective marker *tame*. *ACM Transactions on Asian Language Information Processing*, 4(4):435–474.
- Wei Jin, Hung Hay Ho, and Rohini K. Srihari. 2009. Opinionminer: A novel machine learning system for web opinion mining and extraction. In *Proceedings of the 15th ACM SIGKDD*, pages 1195–1204.

- Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 483–490.
- Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 1–7.
- Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In C.C. Aggarwal and C.X.Zhai, editors, *Mining Text Data*, pages 415–463. Springer.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Journal Foundations and Trends in Information Retrieval*, 2(1–2):1–135.
- Yohei Seki, Noriko Kando, and Masaki Aono. 2009. Multilingual opinion holder identification using author and authority viewpoints. *Journal of the Information Processing and Management*, 45(2):189–199.
- 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*, pages 56–65.

# Cognate Production using Character-based Machine Translation

Lisa Beinborn, Torsten Zesch and Iryna Gurevych

Ubiquitous Knowledge Processing Lab (UKP-TUDA)  
Department of Computer Science, Technische Universität Darmstadt

Ubiquitous Knowledge Processing Lab (UKP-DIPF)  
German Institute for International Educational Research

[www.ukp.tu-darmstadt.de](http://www.ukp.tu-darmstadt.de)

## Abstract

Cognates are words in different languages that are associated with each other by language learners. Thus, cognates are important indicators for the prediction of the perceived difficulty of a text. We introduce a method for automatic cognate production using character-based machine translation. We show that our approach is able to learn production patterns from noisy training data and that it works for a wide range of language pairs. It even works across different alphabets, e.g. we obtain good results on the tested language pairs English-Russian, English-Greek, and English-Farsi. Our method performs significantly better than similarity measures used in previous work on cognates.

## 1 Introduction

In order to improve comprehension of a text in a foreign language, learners use all possible information to make sense of an unknown word. This includes context and domain knowledge, but also knowledge from the mother tongue or any other previously acquired language. Thus, a student is more likely to understand a word if there is a similar word in a language she already knows (Ringbom, 1992). For example, consider the following German sentence:

*Die internationale Konferenz zu kritischen Infrastrukturen im Februar ist eine Top-Adresse für Journalisten.*

Everybody who knows English might grasp the gist of the sentence with the help of associated words like *Konferenz-conference* or *Februar-February*. Such pairs of associated words are called *cognates*.

A strict definition only considers two words as cognates, if they have the same etymological origin, i.e. they are genetic cognates (Crystal, 2011). Language learners usually lack the linguistic background to make this distinction and will use all similar words to facilitate comprehension regardless of the linguistic derivation. For example, the English word *strange* has the Italian correspondent *strano*. The two words have different roots and are therefore genetically unrelated. However, for language learners the similarity is more evident than for example the English-Italian genetic cognate *father-padre*. Therefore, we aim at identifying all words that are sufficiently similar to be associated by a language learner no matter whether they are genetic cognates. As words which are borrowed from another language without any modification (such as *cappuccino*) can be easily identified by direct string comparison, we focus on word pairs that do not have identical spelling.

If the two associated words have the same or a closely related meaning, they are true cognates, while they are called false cognates or false friends in case they have a different meaning. On the one hand, true cognates are instrumental in constructing easily understandable foreign language examples, especially in early stages of language learning. On the other hand, false friends are known to be a source of errors and severe confusion for learners (Carroll, 1992) and need to be practiced more frequently. For these reasons, both types need to be considered when constructing teaching materials. However, existing lists of cognates are usually limited in size and only available for very few language pairs. In order to improve language learning support, we aim at automatically creating lists of related words between two languages, containing both, true and false cognates.



In order to construct such cognate lists, we need to decide whether a word in a source language has a cognate in a target language. If we already have candidate pairs, string similarity measures can be used to distinguish cognates and unrelated pairs (Montalvo et al., 2012; Sepúlveda Torres and Aluisio, 2011; Inkpen et al., 2005; Kondrak and Dorr, 2004). However, these measures do not take the regular production processes into account that can be found for most cognates, e.g. the English suffix *-tion* becomes *-ción* in Spanish like in *nation-nación* or *addition-adición*. Thus, an alternative approach is to manually extract or learn production rules that reflect the regularities (Gomes and Pereira Lopes, 2011; Schulz et al., 2004).

All these methods are based on string alignment and thus cannot be directly applied to language pairs with different alphabets. A possible workaround would be to first transliterate foreign alphabets into Latin, but unambiguous transliteration is only possible for some languages. Methods that rely on the phonetic similarity of words (Kondrak, 2000) require a phonetic transcription that is not always available. Thus, we propose a novel production approach using statistical character-based machine translation in order to directly produce cognates. We argue that this has the following advantages: (i) it captures complex patterns in the same way machine translation captures complex rephrasing of sentences, (ii) it performs better than similarity measures from previous work on cognates, and (iii) it also works for language pairs with different alphabets.

## 2 Character-Based Machine Translation

Our approach relies on statistical phrase-based machine translation (MT). As we are not interested in the translation of phrases, but in the transformation of character sequences from one language into the other, we use words instead of sentences and characters instead of words, as shown in Figure 1. In the example, the English character sequence *cc* is mapped to a single *c* in Spanish and the final *e* becomes *ar*. It is important to note that these mappings only apply in certain contexts. For example, *accident* becomes *accidente* with a double *c* in Spanish and not every word-final *e* is changed into *ar*. In statistical MT, the training process generates a phrase table with transformation probabilities. This information is combined with language model probabilities and a search algo-

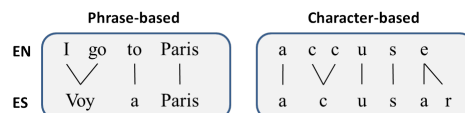


Figure 1: Character-based machine translation

rithm selects the best combination of sequences. The transformation is thus not performed on isolated characters, it also considers the surrounding sequences and can account for context-dependent phenomena. The goal of the approach is to directly produce a cognate in the target language from an input word in another language. Consequently, in the remainder of the paper, we refer to our method as COP (COgnate Production).

Exploiting the orthographic similarity of cognates to improve the alignment of words has already been analyzed as a useful preparation for MT (Tiedemann, 2009; Koehn and Knight, 2002; Ribeiro et al., 2001). As explained above, we approach the phenomenon from the opposite direction and use statistical MT for cognate production.

Previous experiments with character-based MT have been performed for different purposes. Pennell and Liu (2011) expand text message abbreviations into proper English. In Stymne (2011), character-based MT is used for the identification of common spelling errors. Several other approaches also apply MT algorithms for transliteration of named entities to increase the vocabulary coverage (Rama and Gali, 2009; Finch and Sumita, 2008). For transliteration, characters from one alphabet are mapped onto corresponding letters in another alphabet. Cognates follow more complex production patterns. Nakov and Tiedemann (2012) aim at improving MT quality using cognates detected by character-based alignment. They focus on the language pair Macedonian-Bulgarian and use English as a bridge language. As they use cognate identification only as an intermediary step and do not provide evaluation results, we cannot directly compare with their work. To the best of our knowledge, we are the first to use statistical character-based MT for the goal of directly producing cognates.

## 3 Experimental Setup

Figure 2 gives an overview of the COP architecture. We use the existing statistical MT engine Moses (Koehn et al., 2007). The main difference of character-based MT to standard MT is the lim-

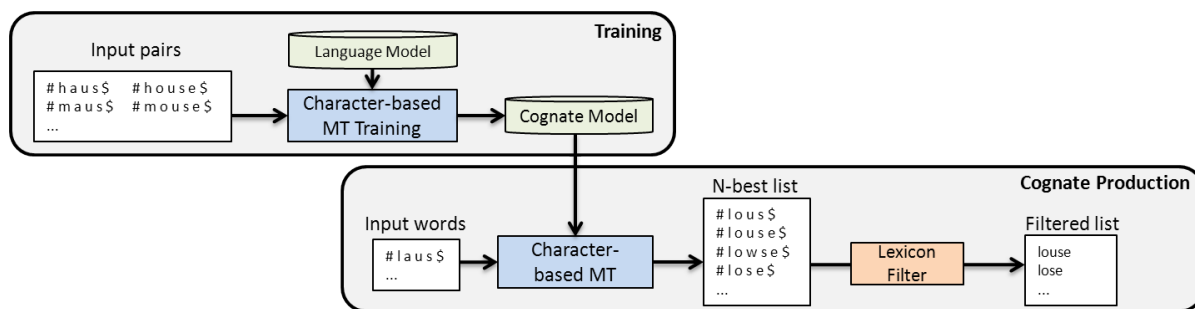


Figure 2: Architecture of our Cognate Production (COP) approach

ited lexicon. Our tokens are character n-grams instead of words, therefore we need much less training data. Additionally, distortion effects can be neglected as reordering of ngrams is not a regular morphological process for cognates.<sup>1</sup> Thus, we deal with less variation than standard MT.

**Training** As training data, we use existing lists of cognates or lists of closely related words and perform some preprocessing steps. All duplicates, multiwords, conjugated forms and all word pairs that are identical in source and target are removed. We lowercase the remaining words and introduce # as start symbol and \$ as end symbol of a word. Then all characters are divided by blanks. Moses additionally requires a language model. We build an SRILM language model (Stolcke, 2002) from a list of words in the target language converted into the right format described above. On the basis of the input data, the Moses training process builds up a phrase table consisting of character sequences in our case. As a result of the training process, we receive a cognate model that can be used to produce cognates in the target language from a list of input test words.

**Cognate Production** Using the learned cognate model, Moses returns a ranked  $n$ -best list containing the  $n$  most probable transformations of each input word. In order to eliminate non-words, we check the  $n$ -best list against a lexicon list of the target language. The filtered list then represents our set of produced cognates. Note that, as discussed in Section 1, the list will contain true and false cognates. The distinction can be performed using a bilingual dictionary (if available) or with statistical and semantic measures for the identification of false friends (Mitkov et al., 2008; Nakov et al., 2007). For language learning, we need both

types of cognates as foreign words also trigger wrong associations in learners (see Section 5.4).

**Evaluation Metrics** In order to estimate the cognate production quality without having to rely on repeated human judgment, we evaluate COP against a list of known cognates. Existing cognate lists only contain pairs of true cognates, but a word might have several true cognates. For example, the Spanish word *música* has at least three English cognates: *music*, *musical*, and *musician*. Therefore, not even a perfect cognate production process will be able to always rank the right true cognate on the top position. In order to account for the issue, we evaluate the coverage using a relaxed metric that counts a positive match if the gold standard cognate is found in the  $n$ -best list of cognate productions. We determined  $n = 5$  to provide a reasonable approximation of the overall coverage.

We additionally calculate the mean reciprocal rank (MRR) as

$$MRR = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{1}{rank_i}$$

where  $C$  is the set of input words and  $rank_i$  the rank of the correct cognate production. For example, if the target cognate is always ranked second-best, then the MRR would be 0.5.<sup>2</sup>

Note that in our language learning scenario, we are also interested in words that might be associated with the foreign word by learners, but are actually not true cognates (e.g. the English word *muse* might also be mistakenly associated with *música* by language learners). Unfortunately, an evaluation of the false cognates produced by COP is not covered by those metrics and thus left to a qualitative analysis as performed in section 5.4.

<sup>1</sup>We use these parameters: -weight-1 1 -weight-d 0 -weight-w -1 -dl 0 -weight-t 0.2 0.2 0.2 0.2 0.2

<sup>2</sup>BLEU (Papineni et al., 2002) is the common evaluation metric for MT, but would be misleading in our setting.

## 4 Experiments & Results

We conducted a set of experiments that cover different aspects of the cognate production process. First, we test whether the approach is able to learn simple production rules. We select optimal parameters and test the influence of the size and quality of the available training data. We then compare our best model to previous work. For these experiments, we use the language pair English-Spanish, as a large manually collected list of cognates is available for training and evaluation.

### 4.1 Ability to Learn Production Rules

We train COP on a list of just ten cognates all following the same production process in order to test whether COP can generally learn cognate production rules. We test two different processes: i) the pattern (*~tion*→*~ción*), as in *tradition-tradición* ii) the pattern (*~ance*→*~ancia*) as in *elegance-elegancia*. The experiment shows that COP correctly produces the respective target cognates for new input words with the same pattern. We can conclude that COP succeeds in learning the necessary patterns for cognate production. In the following, we investigate whether our approach can also be applied to noisy training data containing a mixture of many different production processes.

### 4.2 Parameter Selection

We vary the following COP parameters: the character  $n$ -gram size used for tokenization, the order of the language model, the lexicon used for filtering, and tuning of Moses parameters. We collected a list of 3,403 English-Spanish cognates and split it into training set (2,403), development set (673), and test set (327).<sup>3</sup> Table 1 shows the coverage in the 5 best productions and the MRR for each parameter.

**N-gram Size** We start with the  $n$ -gram size parameter that determines the tokenization of the input, the respective format for unigrams, bigrams, and trigrams for the word *banc* looks as follows:

`# b a n c $ / # b b a a n n c c $ / # b a b a n a n c n c $`

Higher order  $n$ -grams in general increase the vocabulary and thus lead to better alignment. However, they also require a larger amount of training data, otherwise the number of unseen instances is

<sup>3</sup>The cognates have been retrieved from several web resources and merged with the set used by Montalvo et al. (2012). All test cognate list can be found at: <http://www.ukp.tu-darmstadt.de/data>

		Cov. (n=5)	MRR
1)	Unigram	.63	.43
	<b>Bigram</b>	.65	.49
	Trigram	.51	.40
2)	LM-order 5	.68	.48
	<b>LM-order 10</b>	.65	.49
3)	Web1T-Filter	.68	.52
	<b>Wordlist-Filter</b>	.65	.54
4)	Moses Tuning	.66	.54

Table 1: Parameter selection for COP. The settings in bold are used for the subsequent experiments.

too high. We find that bigrams produce slightly better results than unigrams and trigrams, this is in line with findings by Nakov and Tiedemann (2012). Thus, in the following experiments, we use character bigrams.

**Language Model** The next parameter is the language model which determines the probability of a sequence in the target language, e.g. a model of order 5 considers sequences of character  $n$ -grams up to a maximum length of 5. Order 5 seems to be already sufficient for capturing the regular character sequences in a language. However, the ranks for the order-10 model are slightly better and as our “vocabulary” is very limited, we can safely decide for the language model of order 10.

**Lexicon Filter** For filtering the  $n$ -best cognate productions, we tried two different lexicon filter lists. A relatively broad one extracted from the English Web1T (Brants and Franz, 2006) word counts, and a more restrictive corpus-based list. The more restrictive filter decreases the coverage as it also eliminates some correct solutions, but it improves the MRR as non-words are deleted from the  $n$ -best list and the ranking is adjusted accordingly. The choice of the filter adjusts the trade-off between cognate coverage and the quality of the  $n$ -best list. For our language learning scenario, we decide to use the more restrictive filter in order to assure high quality results.

**Moses Parameters** Finally, we tune the Moses parameter weights by applying minimum error rate training (Och and Ney, 2003) using the development set, but it makes almost no difference in this setting. Tuning optimizes the model with respect to the BLEU score. For our data, the BLEU score is quite high for all produced cognate candidates, but it is not indicative of the usefulness of the transformation. A word containing one wrong character is not necessarily better than a word con-

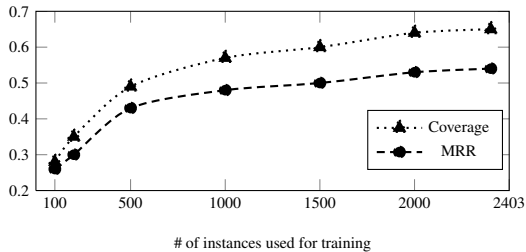


Figure 3: COP learning curve

taining two wrong characters. This explains why tuning has little effect.

Generally, COP reaches a coverage of about 65%. If we consider an n-best list with the 100 best translations (instead of only 5), the coverage increases only by less than 1% on average, i.e. the majority of the correct cognates can be found in the top 5. This is also reflected by the high MRR. In the following experiments, we use the optimal parameter setting (highlighted in Table 1).

### 4.3 Training Data Size & Quality

As we have seen in the experiments in Section 4.1, COP is able to learn a production rule from only few training instances. However, the test dataset contains a variety of cognates following many different production processes. Thus, we evaluate the effect of the size of the training data on COP. The learning curve in Figure 3 shows the results. As expected, both coverage and MRR improve with increasing size of the training data, but we do not see much improvement after about 1,000 training instances. Thus, COP is able to learn stable patterns from relatively few training instances.

However, even a list of 1,000 cognates is a hard constraint for some language pairs. Thus, we test if we can also produce satisfactory results with lower quality sets of training pairs that might be easier to obtain than a list of cognates.

We use word pairs extracted from the freely available multilingual resources UBY (Gurevych et al., 2012) and Universal WordNet (UWN) (de Melo and Weikum, 2009). UBY combines several lexical-semantic resources, we use translations which were extracted from Wiktionary. UWN is based on WordNet and Wikipedia and provides automatically extracted translations for over 200 languages that are a bit noisier compared to UBY translations. Additionally, we queried the Microsoft Bing translation API using all words from

	Training Size	Cov. (n=5)	MRR
Cognates	1,000 / 2,403	.57/.65	.48/.54
Transl.	UBY 1,000 / 6,048	.53/.69	.47/.56
	UWN 1,000 / 10,531	.50/.69	.43/.54
	Bing 1,000 / 5,567	.51/.64	.44/.54
Knowledge-free	1,000 / 34,019	.21/.47	.18/.33

Table 2: Influence of data size and quality

an English word list as query words.<sup>4</sup> We also test a knowledge-free approach by pairing all words from the English and Spanish Web1T corpus.<sup>5</sup> While the translation pairs always share the same meaning, this is not the case for the Web1T pairs, where the majority of pairs will be unrelated.

In order to increase the ratio of possible cognates in the training data, we apply a string similarity filter using the XDICE-measure with a threshold of 0.425<sup>6</sup> on the translation pairs. For the knowledge-free pairs, we use a stricter threshold of 0.6 in order to account for the lower quality.

For a fair quality comparison, we first limit the number of training instances to 1,000, where (as shown above) the performance increases leveled off. The left columns for coverage and MRR in Table 2 show the results. It can be seen, that the results for the translation pairs extracted from UBY, UWN and Bing are only slightly inferior to the use of manually collected cognates for training. The small differences between the resources mirror the different level of linguistic control that has been applied in their creation. The knowledge-free pairs from Web1T yield drastically inferior results. We can conclude that training data consisting of selected cognates is beneficial, but that a high quality list of translations in combination with a string similarity filter can also be sufficient and is usually easier to obtain.

In a follow-up experiment, we use the full size of each training set. As expected, coverage and MRR both increase in all settings. Even with the knowledge-free training set that introduces many noisy pairs, satisfactory results can be obtained. This shows that COP can be used for the production of cognates, even if no language-specific information beyond a lexicon list is available.

### 4.4 Comparison to Previous Work

Previous work (Kondrak and Dorr, 2004; Inkpen et al., 2005; Sepúlveda Torres and Aluisio, 2011;

<sup>4</sup><http://www.bing.com/translator>

<sup>5</sup>We only use every 5th word in order to limit the number of results to a manageable size.

<sup>6</sup>The threshold was selected to cover ~80% of the test set.

	Cov. (n=5)	MRR
DICE	.46	.21
XDICE	.52	.25
LCSR	.51	.24
SpSim	.52	.22
COP	<b>.65</b>	<b>.54</b>

Table 3: Comparison of different approaches for cognate production.

Montalvo et al., 2012) is based on similarity measures that are used to decide whether a candidate word pair is a cognate pair, while COP directly produces a target cognate from the source word. In order to compare those approaches to COP, we pair the English input words from the previous experiments with all words from a list of Spanish words<sup>7</sup> and consider all resulting pairs as candidate pairs. For each pair, we then calculate the similarity score and rank the pairs accordingly. As the similarity measures often assign the same value to several candidate pairs, we get many pairs with tied ranks, which is problematic for computing coverage and MRR. Thus, we randomize pairs within one rank and report averaged results over 10 randomization runs.<sup>8</sup>

We compare COP to three frequently used string similarity measures (LCSR, DICE, and XDICE), which performed well in (Inkpen et al., 2005; Montalvo et al., 2012), and to SpSim which is based on learning production rules. The longest common subsequence ratio (LCSR) calculates the ratio of the length of the longest (not necessarily contiguous) common subsequence and the length of the longer word (Melamed, 1999). DICE (Adamson and Boreham, 1974) measures the shared character bigrams, while the variant XDICE (Brew and McKelvie, 1996) uses extended bigrams, i.e. trigrams without the middle letter. SpSim (Gomes and Pereira Lopes, 2011) is based on string alignment of identical characters for the extraction and generalization of the most frequent cognate patterns. Word pairs that follow these extracted cognate patterns are considered equally similar as pairs with identical spelling.

Table 3 shows the results. The differences between the individual similarity measures are very small, string similarity performs on par with SpSim. The low MRR indicates that the four measures are not strict enough and consider too many candidate pairs as sufficiently similar. COP

<sup>7</sup>In order to ensure a fair comparison, we use the Spanish word list that is also used as lexicon filter in COP.

<sup>8</sup>The average standard deviation is 0.01.

	Language Pair	Cov. (n=5)	MRR
Same alphabet	en-es	.65	.54
	es-en	.68	.48
	en-de	.55	.46
Cross-alphabet	en-ru	.59	.47
	en-el	.61	.37
	en-fa	.71	.54

Table 4: COP results for other languages

performs significantly better than all other measures for both, coverage and MRR. The results for the similarity measures are comparable to the knowledge-free variant of COP (Cov = .47 and MRR = .33, compare Table 2). Obviously, COP better captures the relevant cognate patterns and thus is able to provide a better ranking of the production list. Another advantage of COP is its applicability to language pairs with different alphabets (see Section 5.2), while the similarity measures can only operate within one alphabet.

## 5 Multilinguality

The previous experiments showed that COP works well for the production of Spanish cognates from English source words. However, in language learning, we need to consider all languages previously acquired by a learner, which leads to a large set of language combinations. Imagine, for example, an American physician who wants to learn German. She has studied Spanish in school and the terminology in her professional field has accustomed her to Greek and Latin roots. When facing a foreign text, she might unconsciously activate cues from any of these languages. Thus, if we want to select suitable text for her, we need to consider cognates from many different languages.

In the following experiments, we test how COP performs for other languages with the same alphabet and across alphabets. In addition, we evaluate how well the cognates produced by COP correlate with human judgments.

### 5.1 Same Alphabet

We first analyze whether the cognate production also works in the reverse direction and test the production of English cognates from Spanish source words. The results in Table 4 (upper part) show that COP works bi-directionally, as the scores for Spanish to English are comparable to those for English to Spanish. In addition, we train a model for another Western European language pair, namely English-German. The results show that COP also works well for other language pairs.

English	Spanish	German	Russian	Greek	Farsi
<i>alcohol</i>	alcohol, alcoholar	alkohol, alkoholisch	алкоголь, алкогольный	αλκοολικό, αλκοολικά	الكي, الكل
<i>coffee</i>	café	-	кофей, кофе	-	قهوه
<i>director</i>	director, directora	direktor, direkt	директор	-	غير, دير
<i>machine</i>	machina	maschine, machen	машина, машина	μηχανή, μαχίν	ماشینی, ماشین
<i>music</i>	músico, música	musik, musisch	-	μουσική, μουσικές	موسى, موسيقى
<i>optimal</i>	óptimo	optimal, optimiert	оптимальный	-	مطلوب
<i>popular</i>	popular	populär	популярный	-	محبوب
<i>theory</i>	teoría	theorie	теория	θεωρία, θεωρίας	نظری, تئوری
<i>tradition</i>	tradición	tradition	традиция, традиционный	-	سنتی, سنت

Table 5: Multilingual cognates for English source words produced by COP

## 5.2 Cross-Alphabet

Previous approaches to cognate identification only operate on languages using the same alphabet. As COP is able to learn correspondences between arbitrary symbols, it can easily be applied on cross-alphabet language pairs. In the previous experiments, we had excluded cognate pairs that have exactly the same string representation. For cross-alphabet pairs, this is not possible. Thus, the task is to tackle both, standard transliteration (as in the English-Greek pair *atlas*-άτλας)<sup>9</sup> and cognate production (as in *archangel*-αρχάγγελος)<sup>10</sup>.

We evaluate COP for Russian (ru), Greek (el), and Farsi (fa). For Russian, we use a list of UBY-pairs as training data. Unfortunately, UWN and UBY contain only few examples for Greek and Farsi, so we use Bing translations of English source words. In order to filter the resulting list of words, we transliterate Russian and Greek into the Latin alphabet<sup>11</sup> and apply a string similarity filter. We do not filter the training data for Farsi, as the transliteration is insufficient.

The lower part of Table 4 lists the results. Given that those language pairs are considered to be less related than English-Spanish or English-German, the results are surprisingly good. Especially the production of Farsi cognates works very well, although the training data has not been filtered. The low MRR for Greek indicates that our lexicon filter is not restrictive enough. COP often produces Greek words in several declinations (e.g. nouns in genitive case) which are not eliminated and lead to a worse rank of the correct target. We conclude that COP also works well across alphabets.

## 5.3 Multilingual Cognates

In order to provide the reader with some examples of cognates produced by COP, we compiled a short list of international words that are likely

<sup>9</sup>The transliteration of άτλας is *atlas*.

<sup>10</sup>The transliteration of αρχάγγελος is *ark'aggelos*.

<sup>11</sup>Using ICU: <http://site.icu-project.org/>

to occur in all languages under study. In Table 5, we give the two top-ranked productions. It can be seen that COP produces both, true and false cognates (e.g. *direkt* for *director*), which is useful for language learning scenarios. Of course, some produced forms are questionable, e.g. the second Farsi match for *music* means *Moses*. Note that the gaps in the table are often cases where the absence of a cognate production is an indicator of COP's quality. For example, the Greek words for *director*, *popular*, and *tradition* are not cognates of the English word but have a very different form.

## 5.4 Human Associations

The examples in Table 5 showed that COP produces not only the correct cognate, but all target words that can be created from the input word based on the learned production processes. In order to assess how well these additional productions of COP correlate with human associations, we conducted a user study. We presented Czech words with German origin to 15 native German speakers that did not have any experience with Eastern-European languages. The participants were asked to name up to 3 guesses for the German translation of the Czech source word. Table 6 gives an overview of the Czech source words together with the German associations named by more than one person (number of mentions in brackets). The table shows that some Czech words are strongly associated with their correct German translations (e.g. *nudle*-*Nudel*), while other words trigger false friend associations (e.g. *talíř*-*Taler*).

Another interesting aspect is the influence of languages besides the L1. For example, the German association *himmel* for the Czech word *cíl* is very likely rooted in the Czech-French association

<sup>14</sup>Note that forms like *stak* also pass the lexicon filter, as this is an infrequent, but nevertheless valid German word. Other words like *san* are part of the German lexicon from city names like *San Francisco*.

Czech	Human associations (German)	COP productions (German)
nudle	<b>Nudel</b> (15)	<b>nudel</b> , nadel, ode
švagr	<b>Schwager</b> (13)	sauger, <b>schwager</b> , berg
šlak	<b>Schlag</b> (12), Schlagsahne (3), schlagen (2)	stak
brýle	<b>Brille</b> (12), brüllen (4)	<b>brille</b> , brie
cíl	<b>Ziel</b> (9), Himmel (2)	set, zelle, teller
žold	<b>Sold</b> (9), Zoll (5), Gold (2), verkauft (2), Schuld (2)	<b>sold</b> , gold, geld
sál	Salz (13) , <b>Saal</b> (8)	set, san, all, <b>saal</b>
taška	<b>Tasche</b> (8), Aufgabe (4), Tasse (4), Taste (2)	task, as, tick
skříň	<b>Schrein</b> (5), Bildschirm/Screen (3), schreien (2)	-
flétna	<b>Flöte</b> (4), Flotte (4), Pfannkuchen (2), fliehen (2)	flut, filet
muset	Museum (11), <b>müssen</b> (3), Musik (3), Muse (2), Mus (2)	<b>mus</b> , most, <b>mus</b> , mit
valčík	Walze (4), <b>Walzer</b> (3), falsch (2)	-
talíř	Taler (5), <b>Teller</b> (2), zahlen (3), teilen (2)	<b>teller</b> , <b>taler</b> , ader
šunka	schunkeln (2), Sonne (2), <b>Schinken</b> (1),	sun
knoflík	Knoblauch (11), knifflig (4), <b>Knopf</b> (1)	-

Table 6: Human associations and cognate productions from Czech to German  
Correct translations are in bold, underlined words are COP productions that match human associations.<sup>14</sup>

*cíl-ciel*.<sup>15</sup> A similar process applies for the association *aufgabe*, which is *task* in English and therefore close to *taška*. These cross-linguistic cognitive processes highlight the importance of considering cognates from all languages a learner knows.

In order to examine how well COP reflects the human associations, we train it on manually collected Czech-German cognates and translation pairs from UBY. The number of training instances is rather small, as a language reform in the 19th century eliminated many Czech words with Austrian or German roots. Consequently, the model does not generalize as well as for other language pairs (see the column “COP Productions” in Table 6).<sup>16</sup> However, it correctly identifies cognates like *nudel*, *brille*, and *sold* which are ranked first by the human participants. As we argued above, COP also correctly produces some of the ‘wrong’ associations, e.g. *gold* or *taler*. Thus, COP is to a certain extent able to mimic the association process that humans apply when identifying cognates.

## 6 Conclusions

We introduced COP, a novel method for cognate production using character-based MT. We have shown that COP succeeds in learning the necessary patterns for producing cognates in different languages and alphabets. COP performs significantly better than similarity measures used in previous work on cognates. COP relies on training data, but we have shown that it can be applied even if no language-specific information beyond a word list is available. A user study on German-Czech cognates supports our assumption that COP

productions are comparable to human associations and can be applied for language learning.

In future work, we will focus on the application of cognates in language learning. True cognates are easier to understand for learners and thus can be an important factor for readability assessment and the selection of language learning examples. False cognates, on the other hand, can be confusing and need to be practiced more frequently. They could also be used as good distractors for multiple choice questions. In addition, COP productions that do not pass the lexical filter might serve as pseudo-words in psycholinguistic experiments as they contain very probable character sequences.

## Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Klaus Tschira Foundation under project No. 00.133.2008.

## References

- George W Adamson and Jillian Boreham. 1974. The Use of an Association Measure based on Character Structure to Identify Semantically Related Pairs of Words and Document Titles. *Information Storage and Retrieval*, 10(7):253–260.
- Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1. *Linguistic Data Consortium*.
- Chris Brew and David McKelvie. 1996. Word-Pair Extraction for Lexicography. *Proc. of the 2nd international conference on new methods in language processing*, pages 45–55.
- Susan E. Carroll. 1992. On Cognates. *Second Language Research*, 8(2):93–119, June.

<sup>15</sup>Both words, *himmel* and *ciel* mean *heaven* in English.

<sup>16</sup>Coverage (0.4) and MRR (0.32) are not representative as the test set is too small.

- David Crystal. 2011. *Dictionary of linguistics and phonetics*, volume 30. Wiley-Blackwell.
- Gerard de Melo and Gerhard Weikum. 2009. Towards a Universal Wordnet by Learning from Combined Evidence. *Proc. of the 18th ACM conference*, pages 513–522.
- Andrew Finch and Eiichiro Sumita. 2008. Phrase-based machine transliteration. In *Proc. of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*, pages 13–18.
- Luís Gomes and José Gabriel Pereira Lopes. 2011. Measuring Spelling Similarity for Cognate Identification. *Progress in Artificial Intelligence*, pages 624–633.
- Iryna Gurevych, Judith Ecker-Kohler, Silvana Hartmann, Michael Matuschek, Christian M Meyer, and Christian Wirth. 2012. A Large-Scale Unified Lexical-Semantic Resource Based on LMF. *Proc. of the 13th Conference of the EACL*, pages 580–590.
- Diana Inkpen, Oana Frunza, and Grzegorz Kondrak. 2005. Automatic Identification of Cognates and False Friends in French and English. In *Proc. of the International Conference Recent Advances in NLP*, pages 251–257.
- Philipp Koehn and Kevin Knight. 2002. Learning a Translation Lexicon from Monolingual Corpora. In *Proceedings of the ACL workshop on Unsupervised lexical acquisition*, pages 9–16, July.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Chris Dyer, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics*.
- Grzegorz Kondrak and Bonnie Dorr. 2004. Identification of Confusable Drug Names: A New Approach and Evaluation Methodology. In *Proc. of the 20th international conference on Computational Linguistics*, pages 952–958.
- Grzegorz Kondrak. 2000. A New Algorithm for the Alignment of Phonetic Sequences. In *Proceedings of the 1st NAACL*, pages 288–295.
- I. Dan Melamed. 1999. Bitext Maps and Alignment via Pattern Recognition. *Computational Linguistics*, 25(1):107–130.
- Ruslan Mitkov, Viktor Pekar, Dimitar Blagoev, and Andrea Mulloni. 2008. Methods for extracting and classifying pairs of cognates and false friends. *Machine Translation*, 21(1):29–53, May.
- Soto Montalvo, Eduardo G. Pardo, Raquel Martinez, and Victor Fresno. 2012. Automatic Cognate Identification based on a Fuzzy Combination of String Similarity Measures. *IEEE International Conference on Fuzzy Systems*, pages 1–8, June.
- Preslav Nakov and Jörg Tiedemann. 2012. Combining Word-Level and Character-Level Models for Machine Translation Between Closely-Related Languages. In *Proceedings of the ACL*, pages 301–305.
- Svetlin Nakov, Preslav Nakov, and Elena Paskaleva. 2007. Cognate or False Friend? Ask the Web! In *Proc. of the RANLP workshop: Acquisition and management of multilingual lexicons*, pages 55–62.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of the 40th annual meeting of the ACL*, pages 311–318, July.
- Deana L Pennell and Yang Liu. 2011. A Character-Level Machine Translation Approach for Normalization of SMS Abbreviations. pages 974–982.
- Taraka Rama and Karthik Gali. 2009. Modeling Machine Transliteration as a Phrase Based Statistical Machine Translation Problem. *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, (August):124–127.
- António Ribeiro, Gaël Dias, Gabriel Lopes, and João Mexia. 2001. Cognates Alignment. *Proc. of the Machine Translation Summit 2001*.
- Hakan Ringbom. 1992. On L1 Transfer in L2 Comprehension and L2 Production. *Language Learning*, 42(1):85–112.
- Stefan Schulz, Eduardo Sbrissia, Percy Nohama, and Udo Hahn. 2004. Cognate Mapping - A Heuristic Strategy for the Semi-Supervised Acquisition of a Spanish Lexicon from a Portuguese Seed Lexicon. In *Proc. of the 20th international conference on Computational Linguistics*.
- Lianet Sepúlveda Torres and Sandra Maria Aluisio. 2011. Using Machine Learning Methods to avoid the Pitfall of Cognates and False Friends in Spanish-Portuguese Word Pairs. In *Proc. of the 8th Brazilian Symposium in Information and Human Language Technology*, pages 67–76.
- Andreas Stolcke. 2002. SRILM-an Extensible Language Modeling Toolkit. In *Proc. of the international conference on spoken language processing*, volume 2, pages 901–904.
- Sara Stymne. 2011. Spell Checking Techniques for Replacement of Unknown Words and Data Cleaning for Haitian Creole SMS Translation. *Proc. of the 6th Workshop on SMT*, pages 470–477.
- Jörg Tiedemann. 2009. Character-based PSMT for Closely Related Languages. In *Proc. of 13th Annual Conference of the European Association for Machine Translation*, volume 9, pages 12–19.



# An Empirical Study of Combining Multiple Models in Bengali Question Classification

**Somnath Banerjee**

Department of Computer Science and  
Engineering  
Jadavpur University, India  
s.banerjee1980@gmail.com

**Sivaji Bandyopadhyay**

Department of Computer Science and  
Engineering  
Jadavpur University, India  
sivaji\_cse\_ju@yahoo.com

## Abstract

This paper demonstrates that combination of multiple models achieves better classification performance than that obtained by existing individual models for question classification task in Bengali. We have exploited state of the art multiple model combination techniques, i.e., ensemble, stacking and voting on lexical, syntactical and semantic features of Bengali question for the question classification task. Bagging and boosting have been experimented as ensemble techniques. Naïve Bayes, kernel Naïve Bayes, Rule Induction and Decision Tree classifiers have been used as base learners. The experimental results show that classifier combination models outperform existing single model approaches. Overall voting approach has achieved maximum classification accuracy of 91.65% and outperformed the existing single model approaches (maximum accuracy of 87.63%).

## 1 Introduction

Although different types of question answering systems (QA) have different architectures, most of them follow a framework in which question classification (QC) plays an important role (Voorhees, 2001) and QC has significant influence on the overall performance of a QA system (Ittycheriah et al., 2001; Hovy et al., 2001; Moldovan et al., 2003). The task of a question classifier is to assign one or more class labels, depending on classification strategy, to a given question written in natural language.

Basically there are two main motivations for question classification: locating the answer and choosing the search strategy. Knowing the question class not only reduces the search space needed

to find the answer, it can also help to find the true answer in a given set of candidate answers.

One of the main issues of classification modeling is the improvement of classification accuracy. For that purpose, many researchers have recently placed considerable attention to the task of classifier combination methods. The idea is not to rely on a single decision making scheme. Instead, many single classifiers are used for decision making by combining their individual opinions to arrive at a consensus decision.

## 2 Related Work and Motivations

A lot of researches on QC, question taxonomies, and question features are being published continuously. There are basically two different approaches used to classify questions- one is rule based (Hull, 1999; Prager et al., 1999) and another is machine learning based (Zhang et al., 2003; Li and Roth, 2004). However, a number of researchers have also used some hybrid approaches which combine rule-based and machine learning based approaches (Huang et al., 2008; Silva et al., 2011).

Many researchers have investigated the technique of combining the predictions of multiple classifiers to produce a single classifier (Breiman, 1996; Clemen, 1989; Perrone, 1993; Wolpert, 1992). The resulting classifier is generally more accurate than any of the individual classifiers making up the ensemble. Both theoretical (Hansen and Salamon, 1990; Krogh and Vedelsby, 1995) and empirical (Hashem, 1997; Opitz and Shavlik, 1996a, 1996b) researches have been carried out successfully. Last decade a group of researchers focused on classifier combination methods in question classification task. LI *et al.* (2005) trained four SVM classifiers based on four different types of features and combined them with various strategies. Later LI *et al.* (2006) performed similar type of experiments and achieved

improved accuracy on TREC dataset. (Jia et al., 2007; Su et al., 2009) proposed ensemble learning for Chinese question classification.

Recently, (Banerjee and Bandyopadhyay, 2012) have worked on Bengali QC task and achieved 87.63% accuracy using single classifier approach. So far, classifier combination methods have not been used by any researcher in Bengali question classification task. So, we employ the use of classifier combination methods to improve question classification accuracy.

### 3 Question Type Taxonomies

The present work follows the QC taxonomies proposed by (Banerjee and Bandyopadhyay, 2012) for two reasons. First, that is the only standard taxonomy that exists in Bengali QC so far. Secondly, the results of the present work can be compared with the work of (Banerjee and Bandyopadhyay, 2012) to establish the improvement in accuracy.

### 4 Features

In the task of question classification, there is always an important problem to decide the optimal set of features to train the classifiers. Different studies have extracted various features with different approaches. The features in question classification task can be categorized into three different types: lexical, syntactical and semantic features (Loni, 2011).

Loni *et al.* (2011) also represented a question in the QC task similar to document representation in vector space model, i.e., a question is a vector which is described by the words inside it. Therefore a question  $Q$  can be represented as:

$$Q = (W_1, W_2, W_3, \dots, W_{N-1}, W_N)$$

Where,  $W_K$  = frequency of term  $K$  in question  $Q$ , and  $N$  = total number of Terms

We have also used three types of features for QC. We use the same features previously used by (Banerjee and Bandyopadhyay, 2012).

*Lexical features ( $f_{Lex}$ ):* wh-word, wh-word positions, wh-type, question length, end marker, word shape.

*Syntactical features ( $f_{Syn}$ ):* POS tags, head word.

*Semantic features ( $f_{Sem}$ ):* related words, named entity.

### 5 Combined Model Learning for QC

There are three approaches of classifier combination: 1) Ensemble, 2) Stacking and 3) Voting.

Two popular methods for creating accurate ensembles are *bagging* (Breiman, 1996) and *boosting* (Freund and Schapire, 1996; Schapire, 1990). We have used *Rapid Miner*<sup>1</sup> tool in the experiments of this work.

## 6 Experiments

This section describes our empirical study of *ensemble*, *stacking* and *voting* approaches. Each of these three approaches has been tested with Naïve Bayes (NB), Kernel Naïve Bayes (k-NB), Rule Induction (RI) and Decision Tree (DT). The previous work (Banerjee and Bandyopadhyay, 2012) on Bengali question classification task used these four classifiers. So in this work, we have used those classifiers to establish the effect of combining models.

### 6.1 Dataset

The present research work adopts the same corpus used by (Banerjee and Bandyopadhyay, 2012). The corpus consists of 1100 Bengali questions of different domains, e.g., education, geography, history, science etc. We have used 770 questions (70%) for training and rest 330 questions (30%) to test the classification models.

### 6.2 Results

In total thirteen different experiments have been performed. Four different experiments have been performed for each *bagging* and *boosting*. So, altogether eight different experiments have been performed for the ensemble approach. Four different experiments have been performed for *stacking*. But for *voting*, a single experiment has been performed. Actually, each experiment can be thought of as three experiments, because a classifier model has been tested on  $f_{Lex}$ ,  $f_{Syn} + f_{Sem}$  and  $f_{Lex} + f_{Syn} + f_{Sem}$  features separately. The outcome of the experiments have been tabulated and described in the next sub-sections.

In our study, *classification accuracy* has been used to evaluate the results of the experiments. *accuracy* is the widely used evaluation metric to determine the class discrimination ability of classifiers, and is calculated using the following equation:

$$accuracy(\%) = \frac{T_P + T_N}{P + N}$$

<sup>1</sup><http://www.rapidminer.com>

where,  $T_P$  = true positive samples;  $T_N$  = true negative samples;  $P$  = positive samples;  $N$  = negative samples.

It is a primary metric in evaluating classifier performances and it is defined as the percentage of test samples that are correctly classified by the algorithm.

### 6.2.1 Results based on Bagging

Bagging approach has been applied separately to four classifiers (i.e., NB, k-NB, RI and DT) and Table-1 tabulates the detailed information of the accuracy obtained.

BL	$f_{Lex}$	$f_{Lex}+f_{Syn}$	$f_{Lex}+f_{Syn}+f_{Sem}$
NB	81.53%	82.77%	83.25%
k-NB	82.09%	83.37%	84.22%
RI	83.96%	85.61%	86.90%
DT	85.23%	86.41%	<b>91.27%</b>

Table 1: Experimental results of Bagging.

Initially the size (number of iteration) of the base learner is set to 2. Then experiments have been performed with gradually increased size (size>2). The classification accuracy has been increased with increase in size. But after a certain size, the accuracy has been almost stable. At size=2 and feature= $f_{Lex} + f_{Syn} + f_{Sem}$ , the NB classifier achieves 82.23% accuracy and at size>= 9, it becomes stable with 83.25% accuracy. At size=2 and feature= $f_{Lex} + f_{Syn} + f_{Sem}$ , the k-NB classifier achieves 83.87% accuracy and at size>=15, it becomes stable with 84.22% accuracy. At size=2 and feature= $f_{Lex} + f_{Syn} + f_{Sem}$ , the RI classifier achieves 85.97% accuracy and at size>=8, it becomes stable with 86.90% accuracy. At size=2 and feature= $f_{Lex} + f_{Syn} + f_{Sem}$ , the DT classifier achieves 88.09% accuracy and at size>=7, it becomes stable with 91.27% accuracy. It has been observed from the experiments that at each case Bagging with DT requires less size, i.e., less iteration than the other used classifiers. For experiment with  $f_{Lex}$  features, the bagging size of NB, k-NB, RI and DT are 12, 19, 11 and 10 respectively after which classification accuracy becomes stable. And For experiment with  $f_{Lex} + f_{Syn}$  features, the bagging size of NB, k-NB, RI and DT are 10, 17, 9 and 8 respectively after which classification accuracy becomes stable.

### 6.2.2 Results based on AdaBoost.M1

Like bagging, AdaBoost.M1 has also been applied separately to the four classifiers (i.e., NB, k-NB, RI and DT). Table-2 tabulates the detailed information of the accuracy obtained.

Here, we empirically fix the iterations of AdaBoost.M1 for four classifiers to 12, 16, 10 and 8 respectively for features= $f_{Lex} + f_{Syn} + f_{Sem}$ , because the weight of  $1/\beta_t$  is less than 1 after those values. If  $1/\beta_t$  is less than 1, then the weight of classifier model in boosting may be less than zero for that iteration.

BL	$f_{Lex}$	$f_{Lex}+f_{Syn}$	$f_{Lex}+f_{Syn}+f_{Sem}$
NB	81.74%	82.71%	83.51%
k-NB	83.97%	85.63%	86.87%
RI	83.55%	85.59%	86.27%
DT	85.21%	86.58%	<b>91.13%</b>

Table 2: Experimental results of AdaBoost.M1.

Similarly, for features= $f_{Lex} + f_{Syn}$  and features= $f_{Lex}$  the iterations are 13, 18, 12, 9 and 14, 19, 14, 11 respectively for four classifiers correspondingly. The experiment results show that the performance of k-NB classifier has been improved over RI. But, overall DT performs better than all.

### 6.2.3 Results based on Stacking

In stacking, out of four classifiers three classifiers have been used as the *base learner* (BL) and the remaining classifier has been used as *model learner* (ML). So, four experiments have been conducted separately where each classifier get a chance to be the *model learner*. Table-3 shows the detailed information of the accuracy obtained.

BL	ML	$f_{Lex}$	$f_{Lex}+f_{Syn}$	$f_{Lex}+f_{Syn}+f_{Sem}$
k-NB,RI,DT	NB	81.76%	82.79%	83.64%
NB, RI, DT	k-NB	83.86%	85.54%	86.75%
NB,k-NB,DT	RI	85.55%	87.69%	<b>91.32%</b>
NB,k-NB,RI	DT	85.07%	86.73%	89.13%

Table 3: Experimental results of Stacking.

In the first experiment, three classifiers k-NB, RI and DT have been selected as the *base learners* and the NB classifier has been selected as the *model learner*. Similarly, four experiments have been done selecting k-NB, RI and DT as *model learner* respectively. Experimental results show

that with RI as the *model learner* and NB, k-NB, DT as the *base learners*, the classifier achieves best classification accuracy.

#### 6.2.4 Results Based on Voting

In voting, four classifiers altogether have been used as the *base learners* and *majority vote* has been used as voting approach. Table 4 tabulates the detailed information of the accuracy obtained.

BL	$f_{Lex}$	$f_{Lex}+f_{Syn}$	$f_{Lex}+f_{Syn}+f_{Sem}$
NB, RI, k-NB,DT	86.59%	88.43%	<b>91.65%</b>

Table 4: Experimental results of Voting.

## 7 Conclusions and Perspectives

The automated Bengali question classification system by (Banerjee and Bandyopadhyay, 2012) is based on four classifiers namely Naïve Bayes, Kernel Naïve Bayes, Rule Induction and Decision Tree. Table-5 tabulates the detailed information of the accuracy obtained.

BL	$f_{Lex}$	$f_{Lex}+f_{Syn}$	$f_{Lex}+f_{Syn}+f_{Sem}$
NB	80.65%	81.34%	81.89%
k-NB	81.09%	82.37%	83.21%
RI	83.31%	84.23%	85.57%
DT	84.19%	85.69%	<b>87.63%</b>

Table 5: Experimental results of (Banerjee and Bandyopadhyay, 2012)

Naïve Bayes has been used as the baseline and they have achieved 87.63% accuracy using Decision Tree. But, they have used each classifier as single model separately. The present work shows that classifier combination technique can improve the performance of question classification. Each classifier combination model performs well than single classifier model in terms of classification accuracy.

If we compare the results of previous experiment (Banerjee and Bandyopadhyay, 2012) with *bagging* approach, then classification accuracy of all the classifiers have been notably increased. The classification accuracy on  $f_{Lex}$ ,  $f_{Lex} + f_{Syn}$  and  $f_{Lex} + f_{Syn} + f_{Sem}$  features have been increased by 1.04%, 0.72% and 3.64%. Similarly in the *boosting* approach, the classification accuracy of

all the classifiers have been notably increased and on  $f_{Lex}$ ,  $f_{Lex} + f_{Syn}$  and  $f_{Lex} + f_{Syn} + f_{Sem}$  features the classification accuracy have increased by 1.02%, 0.89% and 3.50%. *Stacking* approach notably increases the accuracy on  $f_{Lex} + f_{Syn}$  features than *bagging* and *boosting* approaches. The classification accuracy on  $f_{Lex}$ ,  $f_{Lex} + f_{Syn}$  and  $f_{Lex} + f_{Syn} + f_{Sem}$  features have been increased by 1.36%, 2.74% and 0.69% respectively. *Voting* approach not only increases the classification accuracy but also hits the maximum accuracy on all features than other combined approaches. *Voting* approach increases the classification accuracy on  $f_{Lex}$ ,  $f_{Lex} + f_{Syn}$  and  $f_{Lex} + f_{Syn} + f_{Sem}$  features by 2.40%, 2.40% and 4.02% respectively.

So, overall *voting* approach with *majority voting* has performed best among all four classifiers combination approaches namely *bagging*, *boosting*, *stacking* and *voting*. Experimental results show that classifiers combination approaches outperform the previous single classifier classification approach by (Banerjee and Bandyopadhyay, 2012) for Bengali question classification.

The main future direction of our research is to exploit other lexical, semantic and syntactic features for Bengali question classification. In future an investigation can be performed on including new Bengali interrogatives using a large corpus. It is also worth investigating fine-grained classes for Bengali questions. In the current work, we have only investigated the Bengali questions. But, this work can be applied to other languages having low resources.

## Acknowledgments

We acknowledge the support of the Department of Electronics and Information Technology (DeitY), Ministry of Communications and Information Technology (MCIT), Government of India funded project “*Development of Cross Lingual Information Access (CLIA) System Phase II*”.

## References

- Abraham Ittycheriah, Franz Martin, Zhu Wei-Jing, Adwait Ratnaparkhi, and Richard J. Mammone. 2001. *IBMs statistical question answering system*. In Proceedings of the 9th Text Retrieval Conference, NIST.
- Anders Krogh and Jesper Vedelsby. 1995. *Neural network ensembles, cross validation, and active learning*. Advances in neural information processing sys-

- tems, Vol. 7, pp. 231-238 Cambridge, MA. MIT Press.
- Babak Loni. 2011. *A survey of state-of-the-art methods on question classification*. Delft University of Technology, Tech. Rep (2011): 1-40.
- Babak Loni, Gijs van Tulder, Pascal Wiggers, Marco Loog, and David Tax. 2011. *Question classification with weighted combination of lexical, syntactical and semantic features*. TSD, pages 243-250.
- Dan Moldovan, Marius Pasca, SandaHarabagiu, and MihaiSurdeanu. 2003. *Performance issues and error analysis in an open-domain question answering system*. ACM Trans. Inf. Syst., 21:133-154.
- David A. Hull. 1999. *Xerox TREC-8 question answering track report*. In Voorhees and Harman.
- David H. Wolpert. 1992. *Stacked generalization*. Neural Networks, 5, 241-259.
- David W. Opitz and Jude W. Shavlik. 1996a. *Actively searching for an effective neural network ensemble*. Connection Science, 8( 3/4): 337-354.
- David W. Opitz and Jude W. Shavlik. 1996b. *Generating accurate and diverse members of a neural network ensemble*. Advances in Neural Information Processing Systems, Vol. 8, pp. 535-541 Cambridge, MA. MIT Press.
- Dell Zhang and Wee Sun Lee. 2003. *Question classification using support vector machines*. ACM SIGIR, pages 26-32, New York, USA, ACM.
- Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin yew Lin, and Deepak Ravichandran. 2001. *Toward semantics-based answer pinpointing*.
- Ellen M. Voorhees. 2001. *Overview of the TREC 2001 question answering track*. TREC, pp. 42-51.
- Joao Silva, Luisa Coheur, Ana Mendes, and Andreas Wichert. 2011. *From symbolic to sub-symbolic information in question classification*. Artificial Intelligence Review, 35(2):137-154.
- John Prager, Dragomir Radev, Eric Brown, and Anni Coden. 1999. *The use of predictive annotation for question answering in trec8*. TREC-8, pp.399-411. NIST.
- Keliang Jia, Kang Chen, Xiaozhong Fan, Yu Zhang. 2007. *Chinese Question Classification Based on Ensemble Learning*. ACIS. pp. 342-347.
- Lars Kai Hansen, and Peter Salamon. 1990. *Neural network ensembles*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 993-1001.
- Lei Su, Hongzhi Liao, Zhengtao Yu, Quan Zhao. 2009. *Ensemble Learning for Question Classification*. ICIS 2009. pp. 501-505.
- Leo Breiman. 1996. *Stacked regressions*. Machine Learning, 24(1), 49-64.
- LI Xin, Xuan-Jing HUANG, and Li-de WU. 2006. *Question Classification by Ensemble Learning*. IJCSNS, 6(3), page : 147.
- Michael Peter Perrone. 1993. *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extension to General Convex Measure Optimization*. Ph.D. thesis, Brown University, Providence, RI.
- Robert E. Schapire. 1990. *The strength of weak learnability*. Machine Learning, 5(2), page:197-227.
- Robert T. Clemen. 1989. *Combining forecasts: A review and annotated bibliography*. International Journal of Forecasting 5, no. 4: 559-583.
- Sherif Hashem. 1997. *Optimal linear combinations of neural networks*. Neural Networks, 10 (4), pp:599-614.
- Somnath Banerjee and Sivaji Bandyopadhyay. 2012. *Bengali Question Classification: Towards Developing QA System*. In Proceedings of SANLP-COLING, pages 25-40, Mumbai, India.
- Somnath Banerjee and Sivaji Bandyopadhyay. 2012a. *Question Classification and Answering from Procedural Text in English*. In Proceedings of QACD-COLING, pages 11-26, Mumbai, India.
- Xin Li and Dan Roth. 2004. *Learning question classifiers: The role of semantic information*. COLING, pp. 556-562.
- Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. *Question classification using head words and their hypernyms*. EMNLP, pp. 927-936.

# A Two-Stage Classifier for Sentiment Analysis

Dai Quoc Nguyen and Dat Quoc Nguyen and Son Bao Pham

Faculty of Information Technology  
University of Engineering and Technology  
Vietnam National University, Hanoi  
{dainq, datnq, sonpb}@vnu.edu.vn

## Abstract

In this paper, we present a study applying reject option to build a two-stage sentiment polarity classification system. We construct a Naive Bayes classifier at the first stage and a Support Vector Machine at the second stage, in which documents rejected at the first stage are forwarded to be classified at the second stage. The obtained accuracies are comparable to other state-of-the-art results. Furthermore, experiments show that our classifier requires less training data while still maintaining reasonable classification accuracy.

## 1 Introduction

The rapid growth of the Web supports human users to easily express their reviews about such entities as products, services, events and their properties as well as to find and evaluate the others' opinions. This brings new challenges for building systems to categorize and understand the sentiments in those reviews.

In particular, document-level sentiment classification systems aim to determine either a positive or negative opinion in a given opinionated document (Turney, 2002; Liu, 2010). In order to construct these systems, classification-based approaches (Pang et al., 2002; Pang and Lee, 2004; Mullen and Collier, 2004; Whitelaw et al., 2005; Kennedy and Inkpen, 2006; Martineau and Finin, 2009; Maas et al., 2011; Tu et al., 2012; Wang and Manning, 2012) utilizing machine learning to automatically identify document-level sentiment polarity are still mainstream methods obtaining state-of-the-art performances. It is because of possibly combining various features such as: bag of words, syntactic and semantic representations as well as exploiting lexicon resources (Wilson et al., 2005; Ng et al., 2006; Taboada et al., 2011) like SentiWordNet (Baccianella et al., 2010). In these systems, Naive Bayes (NB) and Support Vector Machine (SVM) are often applied for training learning models as they are frequently used as baseline methods in task of text classification (Wang and Manning, 2012). Although NBs are very fast classifiers requiring a small amount training data, there is a loss of accuracy due to the NBs' conditional independence assumption. On the other hand, SVMs

achieve state-of-the-art results in various classification tasks; however, they may be slow in the training and testing phases.

In pattern recognition systems, reject option (Chow, 1970; Pudil et al., 1992; Fumera et al., 2000; Fumera et al., 2004) is introduced to improve classification reliability. Although it is very useful to apply reject option in many pattern recognition/classification systems, it has not been considered in a sentiment classification application so far.

In this paper, we introduce a study combining the advantages of both NB and SVM classifiers into a two-stage system by applying reject option for document-level sentiment classification. In the first stage of our system, a NB classifier, which is trained based on a feature representing the difference between numbers of positive and negative sentiment orientation phrases in a document review, deals with easy-to-classify documents. Remaining documents, that are detected as "hard to be correctly classified" by the NB classifier in the use of rejection decision, are forwarded to process in a SVM classifier at the second stage, where the *hard* documents are represented by additional bag-of-words and topic-based features.

## 2 Our approach

This section is to describe our two-stage system for sentiment classification. Figure 1 details an overview of our system's architecture.

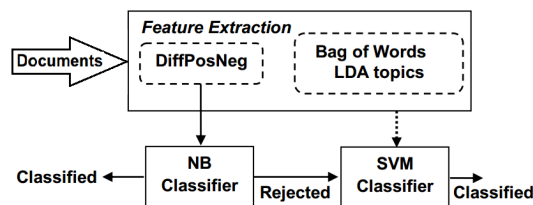


Figure 1: The architecture of our two-stage classifier.

In this positive (pos) and negative (neg) classification problem of sentiment polarity, we reject every sentiment document  $D$  satisfying the following rejection decision based on conditional probabilities:

$$(\tau_1 > P(pos|D) \text{ and } P(pos|D) \geq P(neg|D))$$

**OR**

$$(\tau_2 > P(neg|D) \text{ and } P(neg|D) > P(pos|D))$$

where thresholds  $\tau_1, \tau_2 \in [0, 1]$ . Otherwise, if document  $D$  does not satisfy the rejection decision, it is accepted to be classified by the NB.

A NB classifier at the first stage is to categorize accepted documents. Rejected sentiment documents, that are determined as *hard* to be correctly classified (most likely to be miss-classified) by the NB classifier in applying reject option, are processed at the second stage in a SVM classifier. In our system, the NB classifier categorizes document reviews based on a feature namely DiffPosNeg while the SVM one classifies document reviews with additional bag-of-words (BoW) and topic features.

### DiffPosNeg feature

We exploit the opinion lexicons<sup>1</sup> of positive words and negative words (Hu and Liu, 2004) to detect the sentiment orientation of words in each document. We then employ basic rules presented in (Liu, 2010) to identify the sentiment orientation of phrases. The numerical distance between the numbers of positive and negative opinion phrases in a document  $D$  is referred to as its DiffPosNeg feature value.

### BoW features

The BoW model is the most basic representation model used in sentiment classification, in which each document is represented as a collection of unique unigram words where each word is considered as an independent feature. We calculate the value of feature  $i$  in using *term frequency - inverse document frequency* weighting scheme for the document  $D$  as following:

$$BoW_{iD} = \log(1 + tf_{iD}) * \log \frac{|\{D\}|}{df_i}$$

where  $tf_{iD}$  is the occurrence frequency of word feature  $i$  in document  $D$ ,  $|\{D\}|$  is the total number of documents in the data corpus  $\{D\}$ , and  $df_i$  is the number of documents containing the feature  $i$ . We then normalize BoW feature vector of the document  $D$  as below:

$$\overrightarrow{\eta BoW_D} = \frac{\sum_{\delta \in \{D\}} \|\overrightarrow{BoW_\delta}\|}{|\{D\}| * \|\overrightarrow{BoW_D}\|} * \overrightarrow{BoW_D}$$

### Topic features

Our system also treats each document review as a “bag-of-topics”, and considers each topic as a feature. The topics are determined by using Latent Dirichlet Allocation (LDA) (Blei et al., 2003). LDA is a generative probabilistic model to discover topics for a corpus of documents. LDA represents each document as a probability distribution over latent topics, where each topic is modeled by a probability distribution over words. Using Bayesian inference methods, LDA computes posterior distribution for unseen documents. In our system, we refer to topic probabilities as topic feature values.

<sup>1</sup><http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar>

## 3 Experimental results

### 3.1 Experimental setup

We conducted experiments on the publicly available standard polarity dataset V2.0<sup>2</sup> of 2000 movie reviews constructed by Pang and Lee (2004).

We did not apply stop-word removal, stemming and lemmatization because such stop-words as negation words (e.g: no, not, isn’t) were used in the basic rules to reverse the sentiment orientation of phrases, and as pointed out by Leopold and Kindermann (2002) stemming and lemmatization processes could be detrimental to accuracy. We kept 4000 most frequent words for each polarity class, after removing duplication, we had total 5043 BoW features.

For extracting LDA topic features, we used the JGibbLDA implementation<sup>3</sup> developed by Phan and Nguyen (2007), in which  $\alpha$  is set to 0.5,  $\beta$  is set to 0.1 and the number of Gibbs sampling iterations is set to 3000. We exploited a corpus<sup>4</sup> of 50000 unlabeled movie reviews published by Maas et al. (2011) to build LDA topic models. We then applied these models to compute the posterior probability distribution over latent topics for each movie review in the experimented dataset of 2000 reviews.

In order to compare with other published results, we evaluate our classifier based on 10-fold cross-validation. We randomly separate the dataset into 10 folds; giving one fold size of 100 positive and 100 negative reviews. This evaluation procedure is repeated 10 times that each fold is used as the testing dataset, and 9 remaining folds are merged as the training dataset. All our performance results are reported as the average accuracy over the testing folds.

We utilized WEKA’s implementations (Hall et al., 2009) of NB and SVM’s fast training Sequential Minimal Optimization algorithm (Platt, 1999) for learning classification with the WEKA’s default parameters (e.g: the linear kernel for SVM).

### 3.2 Results without reject option

Table 1 provides accuracies achieved by the single NB and SVM classifiers without the reject option: our NB and SVM classifiers were trained on the whole training dataset of 9 folds according to the above 10-fold cross-validation scheme. We consider BoW model as a baseline, similar to other approaches (Pang and Lee, 2004; Whitelaw et al., 2005; Tu et al., 2012).

In table 1, the accuracy results based on only *Diff-PosNeg* feature are 70.00% for NB and 69.55% for SVM. The highest accuracies in utilizing LDA topics are 78.05% for NB classifier and 85.30% for SVM classifier due to 50 topic features. Besides, the accuracy accounted for SVM at 86.30% over the combination of

<sup>2</sup><http://www.cs.cornell.edu/people/pabo/movie-review-data/>

<sup>3</sup><http://jgibblda.sourceforge.net/>

<sup>4</sup><http://ai.stanford.edu/~amaas/data/sentiment/>

Table 2: Results in applying reject option (8 folds for training), and in other SVM-based methods

$\tau_1$	$\tau_2$	$r_{Pos}$	$r_{Neg}$	NB	SVM	Accuracy
0.79	0.81	0.764	0.987	236 13	1519 232 (tuned thresholds)	<b>87.75</b>
0.82	0.80	0.796	0.990	205 9	1554 232	<b>87.95</b>
1.0	1.0	1.0	1.0	0 0	1752 248	87.60
Pang and Lee (2004)				BoW		87.15
				BoW with minimum cuts		87.20
Whitelaw et al. (2005)				BoW (48314 features)		87.00
				BoW and appraisal groups (49911 features)		90.20
Kennedy and Inkpen (2006)				Contextual valence shifters with 34718 features		86.20
Martineau and Finin (2009)				BoW with smoothed delta IDF		88.10
Maas et al. (2011)				Full model and BoW		87.85
				Full model + additional unlabeled data + BoW		88.90
Tu et al. (2012)				BoW		87.05
				BoW & dependency trees with simple words		88.50
Wang and Manning (2012)				NBSVM-Unigram		87.80
				NBSVM-Bigram		89.45

Table 1: Results without reject option

Features	NB	SVM
BoW (baseline)	73.55	<b>86.05</b>
20 LDA topics	77.55	82.05
30 LDA topics	74.95	79.65
40 LDA topics	76.60	82.15
50 LDA topics	78.05	85.30
60 LDA topics	75.80	83.40
DiffPosNeg	70.00	69.55
DiffPosNeg & BoW	73.50	86.30
DiffPosNeg & 50-LDA	79.35	85.45
BoW & 50-LDA	73.60	87.70
DiffPosNeg & BoW & 50-LDA	73.85	87.70

DiffPosNeg and BoW features is greater than the baseline result of 86.05% with only BoW features. By exploiting a full combination of DiffPosNeg, BOW and 50 LDA topic features, the SVM classifier gains the exceeding accuracy to 87.70%.

### 3.3 Results in applying reject option

In terms of evaluating our two-stage approach, if the fold<sub>*i*th</sub> is selected as the testing dataset, the fold <sub>$(i^{th}+1)\%10$</sub>  will be selected as the development dataset to estimate reject thresholds while both NB and SVM classifiers will be learned from 8 remaining folds. By varying the thresholds' values, we have found the most suitable values  $\tau_1$  of 0.79 and  $\tau_2$  of 0.81 to gain the highest accuracy on the development dataset.

Table 2 presents performances of our sentiment classification system in employing reject option, where the NB classifier was learned based on the DiffPosNeg feature, and the SVM classifier was trained on the full combination of DiffPosNeg, BoW and 50 LDA topic features (total 5094 features). In the table 2,  $r_{Pos}$  and  $r_{Neg}$  are reject rates corresponding with positive label and negative label in the *testing* phase:

$$r_{Pos} = \frac{\text{number of rejected positive reviews}}{1000}$$

$$r_{Neg} = \frac{\text{number of rejected negative reviews}}{1000}$$

$$\text{Overall\_reject\_rate} = \frac{r_{Pos} + r_{Neg}}{2}$$

With the values  $\tau_1$  of 0.79 and  $\tau_2$  of 0.81, our two-stage classifier achieves the result of 87.75% on the testing dataset that as illustrated in table 2, it is comparable with other state-of-the-art SVM-based classification systems, many of which used deeper linguistic features. In total 10 times of cross fold-validation experiments for this accuracy, the NB accepted 249 documents to perform classification and rejected 1751 documents to forward to the SVM. Specifically, the NB correctly classified 236 documents whilst the SVM correctly categorized 1519 documents.

Additionally, in the setup of taking 8 folds for training NB and SVM, and not taking 1 fold of development into account, by directly varying values  $\tau_1$  and  $\tau_2$  on the testing dataset, our system can reach the highest result of 87.95% which is 1.9% and 0.35% higher than the SVM-based baseline result (86.05%) and the accuracy (87.60%) of the single SVM classifier without reject option, respectively.

### 3.4 Results in using less training data

To assess the combination of advantages of NB (requiring small amount of training data) and SVM (high performance in classification tasks), we also carried out experiments of using less training data. In this evaluation, if the fold<sub>*i*</sub> is selected as testing data, the fold <sub>$(i+1)\%10$</sub>  will be selected as training dataset to build the NB classifier. Applying the rejection decision on 8 remaining folds with given reject thresholds, the dataset of rejected documents are used to learn the SVM classifier.

In experiments, the single NB classifier without reject option attains an averaged accuracy of 69.9% that



is approximately equal to the accuracy on 9-fold training dataset at 70% as provided in the table 1. This comes from that our proposed *DiffPosNeg* feature is simple enough to obtain a good NB classifier from small training set. In these experiments, the given thresholds applied in the training phase to learn the SVMs are reused in the testing phase (i.e. the same thresholds for both training and testing phases).

Table 3: Reject option results using less training data

$\tau_1$	$\tau_2$	$r_{Pos}^*$	$r_{Neg}^*$	$r_{Pos}$	$r_{Neg}$	$Acc_S$	Accuracy
0.95	0.63	0.722	0.478	0.722	0.475	84.80	80.55
0.64	0.75	0.483	0.723	0.486	0.729	84.80	82.35
0.72	0.65	0.495	0.496	0.491	0.494	83.75	80.50
0.78	0.69	0.606	0.605	0.609	0.600	84.65	82.30
0.88	0.74	0.764	0.770	0.765	0.770	85.80	84.35
0.97	0.78	0.906	0.905	0.908	0.910	86.65	85.75

Table 3 summaries some reject option-based results taking less training data to learn the SVMs based on the full combination of 5094 features, where  $r_{Pos}^*$  and  $r_{Neg}^*$  are reject rates in the *training* phase, and  $Acc_S$  denotes the accuracy of the single SVM classifier without reject option. With the modest overall reject rate of 0.493 in testing phase, our classifier reached an accuracy of 80.50%, which it outperformed the single NB.

Table 4: Results with SVM trained on *DiffPosNeg* and BoW

$\tau_1$	$\tau_2$	$r_{Pos}^*$	$r_{Neg}^*$	$r_{Pos}$	$r_{Neg}$	$Acc_S$	Accuracy
0.95	0.63	0.722	0.478	0.722	0.475	84.50	80.55
0.64	0.75	0.483	0.723	0.486	0.729	84.00	81.60
0.80	0.68	0.618	0.591	0.622	0.585	83.90	81.05
0.85	0.73	0.726	0.745	0.732	0.753	84.65	83.65
0.97	0.78	0.906	0.905	0.908	0.910	85.70	84.85
0.92	0.80	0.854	0.941	0.861	0.945	85.70	85.35

In other experiments using less training data as presented in table 4, we trained the SVM classifier based on the combination of *DiffPosNeg* and BoW features. For the overall reject rate of 0.903 in testing phase, our system gained a result of 85.35% that is a bit of difference against the accuracy of the single SVM at 85.70%.

Table 3 and table 4 show that our classifier produced reasonable results in comparison with single NB and SVM classifiers without reject option.

### 3.5 Discussion

It is clearly that a different set of features could be used for learning the NB classifier at the first classification stage in our system. However, as mentioned in section 3.4, it is sufficient to have a good NB classifier learned from an unique *DiffPosNeg* feature. Furthermore, an obvious benefit of having the NB based on only one *easy-to-extract* feature is to enhance the efficiency in terms of time used in the document classification process. That is the reason why we applied only the *DiffPosNeg* feature at the first stage.

With regards to the processing time efficiency, it is because there are no recognition time evaluations associated to the other compared systems as well as it is not straightforward to re-implement those systems, hence, the comparison over processing time with the other systems is not crucial to our evaluation. Nevertheless, we believe that our classifier enables to get a fast complete recognition in which time spent to extract features is also taken into accounts, where the majority amount of the classification time is allocated to the feature extraction process.

Considering to feature extraction time, let  $\Gamma_1$  be the time taken to extract *DiffPosNeg* feature and  $\Gamma_2$  be the time spent for extracting other features (i.e. BoW and LDA topic features): our two-stage system then costs ( $\Gamma_1 + overall\_reject\_rate * \Gamma_2$ ) as opposed to ( $\Gamma_1 + \Gamma_2$ ) by the single SVM without reject option. Depending on the overall reject rate, our system could get a significant increase in the complete recognition time while the returned accuracy of our system is promising compared to that of the single SVM classifier.

## 4 Conclusion

In this paper, we described a study combining NB and SVM classifiers to construct a two-stage sentiment polarity system by applying reject option. At the first stage, a NB classifier processes easy-to-classify documents. Hard-to-classify documents, which are identified as most likely to be miss-classified by the first NB classifier in using rejection decision, are forwarded to be categorized in a SVM classifier at the second stage.

The obtained accuracies of our two-stage classifier are comparable with other state-of-the-art SVM-based results. In addition, our classifier outperformed a bag-of-words baseline classifier with a 1.9% absolute improvement in accuracy. Moreover, experiments also point out that our approach is suitable for under-resourced tasks as it takes less training data while still maintaining reasonable classification performance.

## Acknowledgment

The authors would like to thank Prof. Atsuhiko Takasu at the National Institute of Informatics, Tokyo, Japan for his valuable comments and kind support.

## References

- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2200–2204.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March.

- C. Chow. 1970. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theor.*, 16(1):41–46, September.
- Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. 2000. Reject option with multiple thresholds. *Pattern Recognition*, 33(12):2099–2101, December.
- Giorgio Fumera, Ignazio Pillai, and Fabio Roli. 2004. A two-stage classifier with reject option for text categorisation. In *Proceedings of Joint IAPR International Workshops SSPR 2004 and SPR 2004*, volume 3138, pages 771–779.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.
- 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.
- Alistair Kennedy and Diana Inkpen. 2006. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2):110–125.
- Edda Leopold and Jörg Kindermann. 2002. Text categorization with support vector machines. how to represent texts in input space? *Mach. Learn.*, 46(1-3):423–444, March.
- Bing Liu. 2010. Sentiment analysis and subjectivity. In Nitin Indurkha and Fred J Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, pages 1–38.
- 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 - Volume 1*, pages 142–150.
- Justin Martineau and Tim Finin. 2009. Delta tfidf: an improved feature space for sentiment analysis. In *Proceedings of the Third Annual Conference on Weblogs and Social Media*, pages 258–261.
- Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP '04*, pages 412–418.
- Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 271–278.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, pages 79–86.
- Xuan-Hieu Phan and Cam-Tu Nguyen. 2007. GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation (LDA).
- John C. Platt. 1999. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208.
- P Pudil, J Novovicova, S Blaha, and J Kittler. 1992. Multistage pattern recognition with reject option. In *Proceedings 11th IAPR International Conference on Pattern Recognition (ICPR'92)*, pages 92–95.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June.
- Zhaopeng Tu, Yifan He, Jennifer Foster, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012. Identifying high-impact sub-structures for convolution kernels in document-level sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '12, pages 338–343.
- 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 '02, pages 417–424.
- 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 (Volume 2: Short Papers)*, ACL '12, pages 90–94.
- Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management, CIKM '05*, pages 625–631.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05*, pages 347–354.

# Exploiting User Search Sessions for the Semantic Categorization of Question-like Informational Search Queries

**Alejandro Figueroa**

Yahoo! Research Latin America  
Av. Blanco Encalada 2120,  
Santiago, Chile  
afiguero@yahoo-inc.com

**Günter Neumann**

DFKI GmbH  
Stuhlsatzenhausweg 3,  
D-66123 Saarbrücken, Germany  
neumann@dfki.de

## Abstract

This work proposes to semantically classify question-like search queries (e.g., “oil based heel creams”) based on the context yielded by preceding search queries in the same user session. Our novel approach is promising as our initial results show that the classification accuracy improved in congruence with the number of previous queries used to model the question context.

## 1 Introduction

Open question answering (QA), i.e., fully automatic systems that find best answers to natural language questions of any type and domain, is still a challenging research problem. On the other hand side, search engines are getting smarter and smarter in order to fulfill users’ information requests. This motivates users to enter more sophisticated search queries (e.g., more complete questions) rather than few keywords, when they are looking for precise information needs (e.g., answers related to precise problems). This is also experienced by the fact that through search engines, it is likely to exploit answer databases of community based question answering (cQA) systems including Yahoo! Answers, if the search query is close to a QA-system like question. Then matching such a question with those in the cQA database is more likely to recognize plausible cQA paraphrases because of close textual relatedness. Furthermore, as the analysis of our data sources suggests, users often express semantically related series of questions in order to guide the search for better answers, and as such, are already performing interactions with search engines. In a general sense, searching is a sequence of queries in the same user session aimed at satisfying an underlying goal that the user is trying to achieve (Rose and Levinson, 2004).

Thus, we believe that it will be inevitable to further automatize a semantic analysis of search queries within user sessions, i.e., to analyze the semantic relatedness of a series of questions whether they constitute actually a session of semantically related questions entered by the same user.

Our contribution into these directions is the exploration of automatic methods to semantically classify question-like search queries, based on the context provided by preceding search queries in the same user session. An important aspect, tackled in this paper, is whether and how much contextual information extracted from user-specific search query sessions helps to effectively train and apply a model to predict the semantic category of a question-like informational search query (cf. (Broder, 2002; Rose and Levinson, 2004)).

Our method recognizes question-like queries by inspecting their associations with Yahoo! Answers pages via user clicks, providing the additional benefit of linking each query with an entry in the Yahoo! Answers category system. Thus our target semantic labeling set comprises 27 categories including business, environment, health, pets, sports and travel. As a consequence, we are able to completely automatize our approach without the need of manually annotated training material, and to automatically create a huge annotated corpus of semantically labeled question-like search queries. We then consider all search queries of a current session entered before the current labeled one as candidate sources for contextual information, and perform different experiments in order to explore the effect of different contextual window sizes. In a nutshell, our approach finished with 50.96% accuracy by exploiting nine previous search queries as window size.

## 2 Related Work

To the best of our knowledge, our work pioneers the idea of profiting from search sessions

for semantically categorizing question-like informational search queries. Broadly speaking, our study is related to community question answering (cQA) (Zhao et al., 2011), user session analysis (Cao et al., 2009), and closer to web query understanding (Reisinger and Pasca, 2011).

In a broad sense, (Rose and Levinson, 2004) proposed a framework for understanding the underlying goals of user searches. They outlined a taxonomy which its first level models three ends: informational (learn something by reading or viewing), navigational (going to a specific web-site) and resource (obtain videos, maps, etc.).

Later, in a more specific manner, the work of (Yin and Shah, 2010) seeks to understand search queries bearing a particular type of entity (e.g., musician) by classifying their generic user intents (e.g., songs, tickets, lyrics and mp3). They built a taxonomy of search intents by exploiting clustering algorithms, capturing words and phrases that frequently co-occur with entities in user queries, and by examining the click relationships between different intent phrases. Posteriorly, (Xue and Yin, 2011) extended this work by organizing query terms within named entity queries into topics, helping to better the understanding of major search intents about entities. The study of (Cheung and Li, 2012) presented an unsupervised approach to cluster queries with similar intents which, in their work, are patterns consisting of a sequence of semantic concepts or lexical items.

In effect, named entities cooperate on understanding user intents better, however detecting named entities in search queries is a difficult task, because named entities are not in standard form and search queries are typically very short (Guo et al., 2009). Thus, (Du et al., 2010) exploited query sequences in search sessions for dealing the lack of context in short queries, when distinguishing named entities on queries.

Our study focuses on the semantic categorization of question-like search queries, which cover a wide variety of informational queries that do not necessarily bear named entities. In particular, this paper studies the impact of preceding queries in user sessions for tackling the lack of context in this semantic categorization. Our approach is supervised trained with a large set of automatically tagged samples via inspecting click patterns between search queries and Yahoo! answers questions.

### 3 Our Approach

This section presents our automatic corpus acquisition and annotation technique, and later the features utilized by our supervised models.

#### 3.1 Corpus Acquisition

Our corpus is distilled from a commercial search engine query log, more specifically, it considers queries in English submitted in the US from May 2011 to January 2013. We extracted ca. 65 millions full user sessions containing questions by keeping only those sessions connected to Yahoo! Answers via at least one user click. We assume that these clicks signal that, at some point during these sessions, users prompted questions and discovered pertinent information on the clicked Yahoo! Answer pages. Since sessions can cover a large period of time, and thus a wide variety of search needs, we split them into transactions by means of two criteria.

First, we benefited from the time difference that two consecutive queries were sent to the search engine. We used a gap of 300 seconds as session splitter, assuming that longer periods of time indicate that users are likely to have changed their search needs. This size for this temporal cut-off has been popularly used for segmenting query logs (Gayo-avello, 2009). Secondly, conventionally, navigational queries (e.g., “*twitter*”) are prompted by users when they want to reach a particular web-site they bear in mind. As a rule of thumb, most frequent queries in search logs are navigational (Broder, 2002; Rose and Levinson, 2004). Thus we used all search queries having a frequency higher than 1,000 across our session corpus as additional transaction splitters.

Next, in order to study the impact of preceding queries in the session on the tagging of a new submitted question-like search query, we kept only transactions containing at least ten queries, where a user click links the tenth or a later query with Yahoo! Answers, and hence with one of its categories. In other words, we studied the impact of until nine historical queries. In total, this pre-processing gave us 1,098,778 transactions, where 15.87% and 3.41% of them are composed exactly of ten and 20 queries, respectively.

Table 1 shows a transaction consisting of 13 queries. Several ten-element transactions can be derived from one transaction. In this table, two query sequences: 1-10 and 3-12 are acquired,

Number	Search query	Clicked hosts
1	you tube how do i make a heel strap	<b>Beauty &amp; Style</b>
2	cracked heel repair	
3	wraps for cracked heel repair	www.pantryspa.com
4	oil based moisturizer brands	
5	oil based moisturizer cream brands	ezinearticles.com
6	oil based moisturizer cream brands	www.alibaba.com
7	oil based moisturizer heel cream brands	www.amazon.com
8	oil based moisturizer heel cream brands	
9	oil based heel cream	
10	is vaseline considered a oil based moisturizer	<b>Beauty &amp; Style</b>
11	vaseline uses	www.ehow.com
12	is vaseline an oil moisturizer	<b>Beauty &amp; Style</b>
13	goodle	www.google.com

Table 1: A transaction (categories are shown for clicked Yahoo! Answers pages).

since queries ten and twelve are connected to Yahoo! Answers. Overall, we obtained 1,772,696 smaller transactions containing only ten elements, in which the 10th query is related to Yahoo! Answers by means of a user click.

### 3.2 Features

Basically, we took into account several features, which were a) derived from all search queries in the transaction; and b) targeted at inferring categories of preceding queries in the transaction, that is to say expect from the one being classified. In the first group, we have:

- **Bag-of-Words (BoW)** models a search query by their words and their respective frequencies.
- WordNet<sup>1</sup> semantic relations for extending search queries with a) words that include query terms in their the semantic range; and b) words that are included in the semantic range of any query term. The former (SR-A) comprises relations such as hypernyms (e.g., pressure → distress) and holonyms (e.g., professor → staff), while the latter (SR-B) relations like hyponyms (e.g., pressure → oil/gas pressure) and meronyms (e.g., service → supplication).

We only considered elements with an absolute frequency higher than three in the corpus. In the second group, that is attributes extracted exclusively from the window size of until nine search queries, we benefited from:

- **Clicked hosts (CH)** are pairs host/click count corresponding to previously clicked URLs

<sup>1</sup>wordnet.princeton.edu

(see table 1). Note that a search query can be connected not necessarily with only one clicked host, but with many.

- **Category terms in URLs (CTU)** checks as to whether or not any of the terms in any previously clicked URL is a term in any of the categories in the Yahoo! Answers taxonomy. We use simple sign matchings to detect word boundaries within full URLs (e.g., slash, hyphen and underscore). We used lower-case for these matchings.
- **Yahoo! Answers Categories (YAC)** of previously clicked Yahoo! answers pages in the session. In our working example (see table 1), the category “*Beauty & Style*”.
- Similarly to YAC, we add words belonging to categories of previously **clicked Wikipedia pages (WC)**. We used words instead of full category names as many are not standardized.

## 4 Experiments and Results

In our empirical setting, we profited from SVM Multiclass as a multi-class classifier<sup>2</sup> (Crammer and Singer, 2001; Tsochantaridis et al., 2004). In all experiments, we use three-fold cross validation operating on our automatically annotated ten queries transaction corpus, since this collection is relatively large.

As for a **baseline**, we built a centroid vector (CV) for each class, and assign to each testing sample the label pertaining to the best scoring centroid vector afterwards. Here, we also conducted a three-fold cross-validation. Results achieved by this baseline and most SVM configurations indicate that the performance improves in tandem with

<sup>2</sup>svmlight.joachims.org/svm\_multiclass.html

h	CV	SVM BoW	SVM BoW +						
			CH	CTU	YAC	WC	SR-A	SR-B	Combined
0	24.31	30.52	-	-	-	-	35.65	41.09	40.23
1	28.19	34.73	28.62	34.78	36.53	33.48	40.72	43.44	44.06
2	30.45	37.99	27.38	36.61	41.27	36.11	41.43	46.56	45.54
3	31.81	41.13	27.64	41.13	45.04	41.16	43.43	46.79	45.45
4	32.60	42.52	30.92	42.37	47.24	42.42	43.52	47.30	46.49
5	33.21	43.75	33.85	43.75	48.95	43.75	44.84	47.60	47.60
6	33.62	<b>44.60</b>	35.60	<b>44.60</b>	49.14	<b>45.12</b>	44.93	47.90	48.76
7	33.87	43.28	37.90	43.20	49.35	43.22	45.79	47.91	49.62
8	34.07	43.59	38.00	43.70	48.02	44.23	46.18	48.94	50.38
9	<b>34.27</b>	43.69	<b>38.39</b>	43.93	<b>50.02</b>	44.83	<b>46.38</b>	<b>49.39</b>	<b>50.96</b>

Table 2: Classification accuracy (%).  $h$  denotes the window (context) size.

the window size, that is the amount of session context. This comparison also shows that SVM exploits the context more efficiently: it requires a smaller number (6) of previous queries to accomplish a growth from 34.27% to 44.60% accuracy (see table 2). This is a key observation as it is also key to maximize the performance using as few as possible context, since this is not always available, especially when the user session is beginning.

Results reaped by models, that ignore context information ( $h=0$ /"Combined" in table 2), show that features, attempting at discovering semantic hints about the new question-like search query, play a vital role. A combination of SR-A and SR-B improve the accuracy by about 10% (from 30.52% to 40.23%). This sheds light on the reason why the clicked host (CH) property was detrimental as several hosts (e.g., Wikipedia) are ambiguous, in other words, they aim at many potential categories. In fact, using this clicked host attribute the performance drops closer to the baseline.

Conversely, evidence from categories related to previously clicked Wikipedia (WC) links aids in enhancing the accuracy with respect to SVM+BoW (45.12% and  $h=6$ ). This improvement is slight as the amount of clicked Wikipedia links is small with respect to the whole collection. On the other hand, categories of previously clicked Yahoo! Answers pages bettered the performance substantially (50.02% and  $h=9$ ). A reason to this is the fact that we are dealing with question-oriented transactions, and hence clicks to Yahoo! Answers can be more frequent and relevant than clicks to Wikipedia. This finding indicates that specialized click patterns manifest across question-oriented search query transactions.

In light of our outcomes, we can conclude that semantic relations provided by WordNet at the word level are extremely useful. In particu-

lar, our figures show that adding SR-B type relations brought about an increase in accuracy from 30.52% to 41.09% and 49.39% without and with session context information, respectively.

Overall, our session context-aware approach combined (column "Combined" in table 2) with our features aimed at inferring semantic content (i.e., SR-A and SR-B) and query categories (i.e., CTU, WC and YAC) finished best (50.96%). This doubled the centroid vector baseline lacking of contextual information and it substantially improved a naive SVM built on BoW.

On a final note, inspecting the confusion matrix corresponding to the best configuration, we discovered that most recurrent misclassifications are due to categories "*Education & Reference*" and "*Health*", which were perceived as "*Science & Mathematics*". These error rates were (59.89%) and (36.25%), respectively.

## 5 Conclusions and Future Work

This study shows that the context provided by preceding queries in user search sessions improves the semantic labeling of QA-like informational search queries. Our results also point out to the positive contribution of semantically-based features.

As future work, we envision the use of linked data for drawing additional semantic inferences, thus assisting in improving the semantic tagging. Additionally, we envisage the use of sharper session segmentation techniques for identifying question-oriented transactions more accurately.

In principle, it would also be possible to build classifiers for checking as to whether or not a user input is a question-like search query, and for determining their semantic classes by some semantic database (e.g., an ontology). Actually, we also leave this open for future research.

## References

- A. Broder. 2002. A Taxonomy of Web Search. In *SIGIR Forum* 36:3-10.
- Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian tao Sun, Enhong Chen, and Qiang Yang. 2009. Context-aware query classification. In *Research and Development in Information Retrieval*, pages 3–10.
- Jackie Chi Kit Cheung and Xiao Li. 2012. Sequence clustering and labeling for unsupervised query intent discovery. pages 383–392.
- Koby Crammer and Yoram Singer. 2001. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2:265–292.
- J. Du, Z. Zhang, J. Yan, Y. Cui, and Z. Chen. 2010. Using search session context for named entity recognition in query. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR*, pages 765–772.
- Daniel Gayo-avello. 2009. A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, 179:1822–1843.
- Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Research and Development in Information Retrieval*, pages 267–274.
- Joseph Reisinger and Marius Pasca. 2011. Fine-grained class label markup of search queries. In *ACL 2011, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1200–1209.
- D. E. Rose and D. Levinson. 2004. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19.
- Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*.
- Xiaobing Xue and Xiaoxin Yin. 2011. Topic modeling for named entity queries. pages 2009–2012.
- Xiaoxin Yin and Sarthak Shah. 2010. Building taxonomy of web search intents for name entity queries. In *World Wide Web Conference Series*, pages 1001–1010.
- Shiqi Zhao, Haifeng Wang, Chao Li, Ting Liu, and Yi Guan. 2011. Automatically generating questions from queries for community-based question answering. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 929–937.

# Influence of Part-of-Speech and Phrasal Category Universal Tag-set in Tree-to-Tree Translation Models

Francisco Oliveira, Derek F. Wong, Lidia S. Chao, Liang Tian, Liangye He

Department of Computer and Information Science,  
University of Macau, Macao SAR, China  
{olifran, derekfw, lidiasc}@umac.mo,  
{tianliang0123, wutianshui0515}@gmail.com

## Abstract

Tree-to-tree Statistical Machine Translation models require the use of syntactic tree structures of both the source and target side in learning rules to guide the translation process. In order to accomplish the task, available treebanks for different languages are used as the main resources to collect necessary information to handle the translation task. However, since each treebank has its own defined tags, a barrier is inherently created in highlighting alignment relationships at different syntactic levels for different tag-sets. Moreover, these models are typically over constrained. This paper presents a unified tag-set for all languages at Part-of-Speech and Phrasal Category level in tree-to-tree models. Different experiments are conducted to study for its feasibility, efficiency, and translation quality.

## 1 Introduction

The study of Statistical Machine Translation (SMT) (Lopez, 2008) relying on syntactic information has received wide attention in recent years. In particular, syntactic information is being integrated either on the source or target or both side(s) in training translation models for handling the translation task. In *hierarchical* models (Chiang, 2007) that consider syntactic information (Zollmann and Venugopal, 2006), the input sentence is analyzed and translated by synchronous context free grammars (SCFG) hierarchically with extra linguistic information. In *string-to-tree* SMT models (Galley et al., 2004; Zhang et al., 2011), the output of the translation always follows a grammatical syntax of the target language. In *tree-to-string* SMT models (Liu et al., 2006; Wu et al., 2010), source side syntax is used to generate the

translation output. Finally, by considering the syntax of both the source and target languages, *tree-to-tree* SMT models (Zhang et al., 2008; Liu et al., 2009) tend to be the best among the previous models. Basically, all of these models require two extra components: (1) *syntax parsers* (He et al., 2012; Petrov et al., 2006) in obtaining annotated syntax trees for training the models, and (2) *monolingual treebanks* (a detailed list can be found in Petrov et al. (2012)) for training the parsers. Currently, many of them are publicly available through Internet, institutions and data consortiums.

Independently from the method used, although there are many treebanks available, they typically have their own tag-set defined for different languages, ranging from tens to hundreds of tags, which is hard to conduct the research in a multilingual environment. As a consequence, Petrov et al. (2012) developed a universal Part-of-Speech (POS) tag-set for twenty five different languages. However, at phrasal level, disagreements between the languages remain undefined.

This paper presents a study of the application of universal tag-set from POS to phrasal category level in tree-to-tree translation models. In the POS tag level, we basically used the universal tag-set proposed by Petrov et al. (2012) in mapping original tags into universal ones. In order to fulfill the missing relationships at phrasal category level, a mapping work of phrasal tags for Chinese (Zh), English (En), French (Fr), German (De), and Portuguese (Pt) is presented. The main objective is to partially relax syntactic constraints imposed to the original models by having more generalizations in the unified tag-set proposed. With fewer tags defined between languages, fewer syntax rules will be extracted during the training phase, which reduces the computation load, possible rule ambiguities, and increases the translation efficiency. Although we only focus on five languages, extensions to other languages are possible.



Tag	Chinese	English	French	German	Portuguese
CNP	CLP, NP, QP, UCP	NP, NAC, NX, WHNP, QP	NP	CNP, MPN, NM, NP	np
CVP	VP, VCD, VCP, VNV, VPT, VRD, VSB	VP	VN, VP, VPpart, VPinf	CVP, VP, VZ	x, vp
CAJP	ADJP	ADJP, WHADJP	AP	AA, AP, CAP, MTA	ap, adjp
CAVP	ADVP, DNP, DP, LCP	ADVP, WHADVP, PRT	AdP	AVP, CAVP	advp
CPP	PP	PP, WHPP	PP	CAC, CPP, PP	pp
CS	FRAG, IP	S, SBAR, SBARQ, SINV, SQ, PRN, FRAG, RRC	ROOT, SENT, Ssub, Sint, Srel	CS, PSEUDO, S	fcl, icl, acl, cu, sq
CCONJP	CP	CONJP	<i>No mapping tag</i>	<i>No mapping tag</i>	<i>No mapping tag</i>
CCOP	<i>No mapping tag</i>	UCP	CCOP	CCP, CO	<i>No mapping tag</i>
CX	LST, PRN	X, INTJ, LST	<i>No mapping tag</i>	CH, CVZ, DL, ISU, QL	<i>No mapping tag</i>

Table 1: Mappings from original Phrasal Category to Universal tags

This paper is organized as follows. Section 2 gives the mapping details from POS and phrasal category level tags into universal ones. Section 3 presents the application of universal tags in tree-to-tree models. Section 4 details the experiment results conducted. Section 5 introduces related work followed by a conclusion.

## 2 Universal Tag-set

A two level universal tag-set is defined in the annotation of syntactic trees for different languages. In the first level, a universal POS tag-set (Petrov et al., 2012) is converted for all leave nodes. It consists of twelve different tags, including: *NOUN* (noun), *VERB* (verb), *ADJ* (adjective), *ADV* (adverb), *PRON* (pronoun), *DET* (determiner and article), *ADP* (preposition and postposition), *NUM* (numeral), *CONJ* (conjunction), *PRT* (particle), “.” (punctuation marks) and *X* (others). However, some tags proposed in their original work are not considered at this stage. For example, the original tag *NP* in English, which is supposed to be converted into *NOUN* at POS level, is only changed to *CNP* at the phrasal category stage for better differentiating its actual meaning at tree level.

In phrasal category level, nine universal tags are defined for higher level nodes: *CNP* (noun phrase), *CVP* (verb phrase), *CAJP* (adjective phrase), *CAVP* (adverb phrase), *CPP* (preposition phrase), *CS* (sentence/sub-sentence), *CCONJP* (conjunction phrase), *CCOP* (coordinated phrase), and *CX* (others). Corresponding mappings at a

phrasal category level for Zh, En, Fr, De, and Pt language are listed in Table 1.

The proposed conversion is carefully designed by studying the actual meaning of the original tags based on previously published work. Although it is common to find out disagreements between tag-sets across different languages due to their inherent characteristics, the objective of this paper is to unify different tags which are used in most of the treebanks at clause level.

## 3 Rule Extraction Process

The rule extraction process for tree-to-tree models based on universal tag-set is similar to hierarchical phrase-based model (Chiang, 2007), which considers SCFG rules for handling the translation task. The main difference is that rules where there are syntactic labels for non-terminals are extracted. Given a word aligned sentence tree pair  $T(f_1^J)$  and  $T(e_1^I)$ , each rule in the model is a three tuple consisting of variables  $ST(f_{j_1}^{j_2})$ ,  $ST(e_{i_1}^{i_2})$ , and  $\tilde{A}$  respectively.  $ST(f_{j_1}^{j_2})$  is a sub-tree covering the interval span  $[j_1, j_2]$  of  $T(f_1^J)$ ; similarly,  $ST(e_{i_1}^{i_2})$  denotes the target sub-tree covering the interval span  $[i_1, i_2]$  of  $T(e_1^I)$ ; and  $\tilde{A}$  is the alignment between terminals and leaf non-terminals of the two trees, such that  $\forall (j, i) \in \tilde{A} : j_1 \leq j \leq j_2 \leftrightarrow i_1 \leq i \leq i_2$  holds.

The extraction process starts with standard phrase extraction, and for all the phrases found, a rule is created for each instance. Based on this initial rule set, the rest of all possible rules are iden-

tified based on a simple criterion: these phrases should be subsumed by larger pairs in this set. As an example, if there is another rule  $\langle ST(\gamma) \parallel ST(\alpha) \parallel A^t \rangle$  such that the pair  $(\gamma, \alpha)$  includes another sub-phrase  $(f_{j_1}^{j_2'}, e_{i_1}^{i_2'})$ , i.e.  $\gamma = \gamma_1 f_{j_1}^{j_2'} \gamma_2$  and  $\alpha = \alpha_1 e_{i_1}^{i_2'} \alpha_2$ , then a new rule  $\langle ST(\gamma_1 X \gamma_2) \parallel ST(\alpha_1 X \alpha_2) \parallel \hat{A} \rangle$  will be created, where  $\hat{A}$  contains alignment information for all the terminals and non-terminals. As syntax information is provided for both sides, for each pair, it must have a node in both trees which subsumes the corresponding string. In other words, non-terminal label checks to their related syntax nodes are necessary in assigning correct tags to all non-terminals in the rules.

$$\begin{aligned}
 & [NP][NP] \text{ 在 巴黎 } [VRD][VBD] \circ [IP] \parallel \\
 & [NP][NP] [VRD][VBD] \text{ in Paris } . [S] \parallel \quad (1) \\
 & 0-0 \ 1-2 \ 2-3 \ 3-1 \ 4-4
 \end{aligned}$$

As an example, in rule (1), the top node of the source tree is  $[IP]$ , the top node of the target tree is  $[S]$ , and both trees have five children. Alignment information between terminals and non-terminals is associated by their numerical positions. It might appear cases in which the source and target node have different tags assigned due to language divergences. As an example, in order to have a valid substitution of  $[VRD][VBD]$ , it requires to have a rule in which the source has a  $VRD$  tag and the target has a  $VBD$  tag. Thus, for all non-terminals except the top node, it consists of the source and target tag.

Once all the rules are learned from the entire corpus, probability scores are calculated, which are used in the decoding stage. In addition, glue rules are added in allowing combinations of partial translation fragments monotonically.

The proposed mapping from the original into universal tag-set is advantageous in two aspects. Firstly, in some sense, after the conversion is performed, some rules become more generalized and relaxed compared to the original model. As an example, in Chinese tag-set, as verb phrase related tags ( $VP$ ,  $VCD$ ,  $VCP$ ,  $VNV$ ,  $VPT$ ,  $VRD$ ,  $VSB$ ) are all grouped into  $CVP$ , more coverage in the selection of rules is expected. In particular, suppose that in the original tag-set, “想一想” (think) is tagged as  $VCD$  (verb compounds), while in universal tag-set, it is tagged as  $CVP$ . In this case, it

is obvious that the phrase “想一想” (think) can only be associated to rules with  $VCD$  but not to verb phrases ( $VP$ ), which limits its usage. As a consequence, a wider coverage of rules is available during the decoding process.

Secondly, since many similar tags in the original tag-set are grouped as only one universal tag, many rules will be merged together, resulting in a smaller size compared to the original model.

## 4 Experiments

The training environment is executed in a server equipped with a Xeon processor at 2.9GHz, with 192G physical memory. All the experiments are carried out in Moses toolkit (Koehn et al., 2007). Different language pairs are considered in the experiments, including Fr-En, De-En, Zh-En, and Zh-Pt. The bilingual data we used for Fr-En and De-En are extracted from Europarl Parliament (version 7), while Zh-En and Zh-Pt parallel information are extracted from online web-sites. All sentences are parsed by Berkeley parser (Petrov et al., 2006) and word-aligned by using GIZA++ based on five iterations of IBM model 1, three for IBM models 3 and 4, and five for HMM alignment (Och and Ney, 2003). We used a 5-gram language model for all the languages based on the SRILM toolkit (Stolcke, 2002).

Different test sets are considered, including: news-test (NT) data (2009, 2010, 2011) for Fr-En and De-En, which are extracted from the international workshop of SMT (WMT) held annually by the ACL’s special interest group for MT; test data for Zh-En and Zh-Pt are extracted from online web pages.

We limited the length of the sentences to be less than fifty, and all of them should be valid aligned parse trees for all the training and testing data. For Chinese, a segmentation model (Zhang et al., 2003) is used for detecting word boundaries.

Table 2 shows the translation quality measured in terms of BLEU metric (Papineni et al., 2002) with the original (Ori.) and universal (Uni.) tag-set. When Chinese is considered as the source, results are lower than the ones targeted for European languages, probably affected by the corpus selection, size of the corpus, parsing success rate, non-standard linguistic phenomena (Wong et al., 2012), etc. In particular, we observed that the parsing accuracy (either on the original or universal tag-set) for Chinese language is lower compared

	Fr-En		De-En	
	Ori.	Uni.	Ori.	Uni.
NT 2009	11.57	11.59*	9.64	9.66
NT 2010	10.81	10.84*	10.48	10.55*
NT 2011	12.12	12.15	9.43	9.44
	Zh-En		Zh-Pt	
	Ori.	Uni.	Ori.	Uni.
Test Data	4.79	4.85	3.87	3.88

Table 2: Translation quality comparison

Language Pair	System	VmPeak (KB)	Rule Size
Fr-En	Ori.	1,002,040	1,223,261
	Uni.	982,208	1,190,177
De-En	Ori.	761,108	926,907
	Uni.	745,724	887,317
Zh-En	Ori.	826,032	853,315
	Uni.	812,144	832,099
Zh-Pt	Ori.	686,932	813,405
	Uni.	682,308	804,356

Table 3: Memory usage and rule size

with other languages, which possibly led to poorer alignment relationships at tree level. However, there is an improvement for all the language pairs with different test sets considered by comparing with the baseline approach. Moreover, we measured the improvements over the baseline based on the significant test method proposed by Koehn (2004). The results that are significantly better than the baseline at  $p = 0.05$  are shown by \*. For NT 2010, the results are totally significant, while others' significance rate is better at a range between 97% and 99.4%.

Table 3 measures the average peak virtual memory (VmPeak) usage, and the actual number of rules generated. It is concluded that there is a decrease of 2% in terms of the peak virtual memory compared to the baseline, and a decrease of 1% to 4% in terms of distinct rules.

In short, although the improvement in terms of the translation quality is not high, it significantly reduces not only the rule table size but also memory requirements, which is very beneficial when larger data are considered.

## 5 Related Work

Some of earlier work focused in describing alignment relationships in dependency tree-to-tree

structures based on synchronous tree mapping grammars (Eisner, 2003), and synchronous dependency insertion grammars (Ding and Palmer, 2005). However, their work is targeted on dependency grammars, which is simpler than CFG equivalent formalisms (Fox, 2002). Other studies reported the use of syntactic information from conventional bilingual parsed trees. Zhang et al. (2008) proposed a tree sequence alignment model for bilingual trees. Liu et al. (2009) considered packed forests instead of 1-best trees for the whole translation process. Although both methods tend to increase rule coverage and to relax the over-constrained problem, they require tailored and sophisticated decoders. Zhai et al. (2011) considered the addition of bilingual phrases and binarization of parse trees to deal with the problems.

In this work, we proposed the substitution of original tags into universal ones, which has a higher level of abstraction in partially increasing the rule coverage while reducing the size of the rule table. Moreover, our approach does not require big changes in tree-to-tree models for accomplishing the translation task.

## 6 Conclusion

This paper presents the application of universal tag-set defined at the POS and the phrasal category level to tree-to-tree models. A phrasal category tag-set is defined for Chinese, English, French, German, and Portuguese. With the universal tag-set, learned rules become more generalized and compact. Moreover, this could partially relax the over-constrained disadvantage of traditional tree-to-tree models. Based on the experiment results, better accuracy is obtained compared with the baseline (without tag conversion) and better efficiency due to the reduced number of rules in the proposed method. In the future, we intend to further evaluate the proposed strategy for more languages, with proper universal tags defined, and to study their actual relationships in the learned rules in deducing new strategies to further reduce the rule table size.

## Acknowledgments

This work is supported by the Research Committee of University of Macau and Science and Technology Development Fund of Macau under the grants UL019B/09-Y3/EEE/LYP01/FST and 057/2009/A2, respectively.

## References

- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548.
- Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, pages 205–208.
- Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 304–311.
- Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proceedings of the North American Chapter of the ACL: Human Language Technologies*, pages 273–280.
- Liangye He, Derek F. Wong, and Lidia S. Chao. 2012. Adapting multilingual parsing models to sinica tree-bank. In *Proceedings of the 2nd CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 211–215.
- 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.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 388–395.
- Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 609–616.
- Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proceedings of the 4th International Joint Conference on Natural Language Processing*, pages 558–566.
- Adam Lopez. 2008. Statistical machine translation. *ACM Computing Survey*, 40(3):1–49.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 433–440.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 2089–2096.
- Andreas Stolcke. 2002. Srilm—an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 901–904.
- Fai Wong, Francisco Oliveira, and Yiping Li. 2012. Hybrid machine aided translation system based on constraint synchronous grammar and translation corresponding tree. *Journal of Computers*, 7(2):309–316.
- Xianchao Wu, Takuya Matsuzaki, and Jun’ichi Tsujii. 2010. Improve syntax-based translation using deep syntactic structures. *Machine Translation*, 24(2):141–157.
- Feifei Zhai, Jiajun Zhang, Yu Zhou, and Zong Chengqing. 2011. Simple but effective approaches to improving tree-to-tree model. In *Proceedings of MT Summit XIII*, pages 261–268.
- Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. Hhmm-based chinese lexical analyzer ictclas. In *Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17*, pages 184–187.
- Min Zhang, Hongfei Jiang, AiTi Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 559–567.
- Jiajun Zhang, Feifei Zhai, and Chengqing Zong. 2011. Augmenting string-to-tree translation models with fuzzy use of source-side syntax. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 204–215.
- Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141.

# Interest Analysis using PageRank and Social Interaction Content

**Chung-chi Huang**

Institute of Information Science  
Academia Sinica  
Taipei, Taiwan 115  
u901571@gmail.com

**Lun-Wei Ku**

Institute of Information Science  
Academia Sinica  
Taipei, Taiwan 115  
lwku@iis.sinica.edu.tw

## Abstract

We introduce a method for learning to predict reader interest. In our approach, social interaction content and both syntactic and semantic features of words are utilized. The proposed method involves estimating topical interest preferences and determining the informativity between articles and their social content. In interest prediction, we integrate articles' *quality* social feedback representing readers' opinions into articles to get information which may identify readers' interests. In addition, semantic aware PageRank is used to find reader interest with the help of word interestingness scores. Evaluations show that PageRank benefits from proposed features and interest preferences inferred across articles. Moreover, results conclude that social interaction content and the proposed selection process help to accurately cover more span of reader interest.

## 1 Introduction

Web keyword extraction tools such as KEA ([www.nzdl.org/Kea/](http://www.nzdl.org/Kea/)) typically look at articles from authors' perspective to calculate the importance of a word in articles. However, keywords are not necessarily words that interest readers. We found that articles could be analyzed more towards reader interest if a system exploited social interaction content (e.g., reader feedback) in social media.

Consider the content of an example article. The Web post describes a newly-renovated old house and the history, life style, and surrounding sightseeing sites of a historical city where it is located. Most keyword tools can easily identify keywords *the old house* (謝宅) and *the historical city* (台南). However, article readers might also be interested in less frequent words like *life style* (生活) and *traditional market* (市場), and single-occurrence like *rental fees* (費用), which are also mentioned in most reader feedback.

In the proposed method, an article was transformed into a word graph where vertices were words in the article and edges between vertices indicated words' co-occurrences. To distinguish associate/key words from words of reader interest, readers' *quality* interaction feedback was considered when building the word graph. Subsequently, word interest preferences and PageRank were utilized to find interest terms. Weightings concerning syntactic and semantic features are utilized in PageRank. Moreover, content-source and content-word weighted PageRank were exploited to return words for interest evaluation. The predicted interests can further be used as candidates for social tagging or article recommendation.

## 2 Related Work

The state-of-the-art keyword extraction methods have been applied to a myriad of natural language processing tasks including document categorization and summarization (Manning and Schütze, 2000; Litvak and Last, 2008), indexing (Li et al., 2004), information retrieval (Turney, 2000), and text mining on social networking or micro-blogging services (Li et al., 2010; Zhao et al., 2011; Wu et al., 2010). Here we extract keywords related to readers' interests.

Recently, collaborative tagging or social tagging has grown in popularity among Web services and received much attention (Golder and Huberman 2006; Halpin et al., 2007). Instead of analyzing user (tagging) activity or tag frequencies, we analyze articles and their social interaction content to predict reader interests.

Researches have been done on reader profiling for content recommendation. White et al. (2009) examined five types of contextual information in website recommendation while Ye *et al.* (2012) further explored social influence on item recommendation. Moreover, Tsagkias and Blanco (2012) concentrated on analyzing users' browsing behavior on news articles, and Jin (2012) recommended contents through a unified, per-

sonalized messaging system. In our work, the accumulated social interaction content is utilized to help determine the interest of future reader.

In studies more related to our work, Liu (2010) and Zhao (2011) present PageRank for keyword analyses using article topic information. The main difference from our current work is that we integrate social content and (global) topical interest preferences for words into (local) content-word weighted PageRank algorithm.

### 3 Finding Interests

To introduce the finding process of interests, we start from the problem statement. Given an article collection of various topics from social media (e.g., blogs), an article *ART*, and its reader feedback *FB*, our goal is to determine a set of interest words that are likely to represent the interest of future readers after reading *ART*.

#### 3.1 Estimating Topical Interest Preferences

Basically, the estimation of topical interest preferences is to calculate the significance or degree of references of a word in a domain topic. The learning process contains four stages: (1) Generate article-word pairs in training data, (2) Generate topic-word pairs in training data, (3) Estimate interest preferences for words w.r.t. article topics based on different strategies, and (4) Output word-and-interest-preference-score pairs for various estimation strategies. In the first two stages of the learning process, we generate two sets of article and word information. The input to these stages is a set of articles with author-chosen topics and, if any, their reader feedback responses. The output is a set of pairs of article ID and word in the article, e.g., (*art*=1, *w*="old house"), and a set of pairs of article topic and word in the article, e.g., (*tp*="travel", *w*="old house"). Note that the article referred here may or may not contain the social reader feedback (See Section 4). In the third stage, we utilize aforementioned sets to estimate reader interest preferences for words across articles and across domain topics. Six different estimation strategies are as follows.

**tfidf.** The first estimation is a traditional yet powerful one, tfidf (term frequency multiplied by inversed document frequency):

$$\text{tfidf}(art, w) = \text{freq}(art, w) / \text{artFreq}(w).$$

**Pr(*w*|*tp*).** The second leverages a word's Maximum Likelihood Estimation under a given topic:

$$\Pr(w | tp) = \text{freq}(tp, w) / \sum_{w'} \text{freq}(tp, w').$$

**Pr(*tp*|*w*).** The third computes the topic-wise senses of a word:

$$\Pr(tp | w) = \text{freq}(tp, w) / \sum_{tp'} \text{freq}(tp', w).$$

**entropy.** The fourth is entropy which utilizes the uncertainty in topics to estimate its topic spectrum or its topic focus:

$$\text{entropy}(w) = -\sum_{tp} \Pr(tp | w) \times \lg(\Pr(tp | w)).$$

**Pr-Entropy(*w*|*tp*).** The fifth further considers topic uncertainty in MLE:  $\Pr(w | tp) / 2^{\text{entropy}(w)}$ .

**Pr-Entropy(*tp*|*w*).** The last is a combination of the third and the fourth:  $\Pr(tp | w) / 2^{\text{entropy}(w)}$ .

These six estimations all take global information (i.e., article collection) into account.

#### 3.2 Predicting Interest for Future Reader

Reader interests were predicted using the procedure in Figure 1. In this procedure we exploit semantic aware PageRank and reader feedback in social media to evaluate readers' interest in an article word. According to our observations, the collection of the reader feedback may reveal the common interest and browse habits of potential readers of the same article.

However, not all reader feedback responds to the article. Therefore, we screen reader feedbacks in Step (1) based on the article *ART*, its feedbacks *FB* and interest preference scores *IntPrefs*. The algorithm for identifying reader responses of a good quality, called *quality* reader responses hereafter, is as follows.

(1) *ngramsart* = generateNgram(*ART*)

(2) *Focused* = findFocused(*IntPrefs*)

(3) *selectedSt* = NULL

for each sentence *st* in *FB*

(4a) *ngramsst* = generateNgram(*st*)

(4b) *informativityco* = Coverage-evaluate(*ngramsst*, *ngramsart*)

(4c) *informativityfo* = Focus-evaluate(*ngramsst*, *Focused*)

(4d) append *st* into *selectedSt* if conditions hold

return *selectedSt*

Each response is evaluated at sentence level concerning informativity checked in two aspects. The first concerns the topic cohesion between reader response sentence *st* and article *ART*. Similar to BLEU's (Papineni et al., 2002) weighted ngram precision in machine translation, we compute the weighted ngram coverage of *st* (Step (4b)) on *ART* and favor the coverage of longer ngrams. Larger ngram coverage indicates higher topic correlation between the two. The second considers the topic distributions of words in *st*. We first rank and identify the words expected to have low topic uncertainty. Entropy estimation in

Section 3.2 is used for this purpose to find *Focused* (Step (2)). Then the informativity on topic focus of *st* is computed as the percentage of its words in set *Focused*. In the end, we prune reader sentences in *FB* according to the thresholds set for *informativity<sub>co</sub>* and *informativity<sub>fo</sub>* (Step (4d)).

After incorporating quality feedback *qualityFB* into *ART* (Step (2) in Figure 1), we construct a word graph for both the article and social content. The word graph is represented by a *v*-by-*v* matrix **EW** where *v* is the vocabulary size. **EW** stores normalized edge weights for word  $w_i$  and  $w_j$  (Step (4) and (5)). Note that the graph is directional from  $w_i$  to  $w_j$  and that edge weights are the words' co-occurrence counts within window size *WS*.

```

procedure PredictInterest(ART,FB,IntPrefs,λ,α,N)
(1) qualityFB=selectInformativeFB(ART,FB,IntPrefs)
(2) Concatenate ART with qualityFB into Content
//Construct word graph for PageRank
(3) EWv×v=0v×v
    for each sentence st in Content
        for each word  $w_i$  in st
            for each word  $w_j$  in st where  $i < j$  and  $j - i \leq WS$ 
                if not IsContWord( $w_i$ ) and IsContWord( $w_j$ )
(4a)    EW[i,j]+= $1 \times m \times srcWeight$ 
                elif not IsContWord( $w_i$ ) and not IsContWord( $w_j$ )
(4b)    EW[i,j]+= $1 \times (1/m) \times srcWeight$ 
                elif IsContWord( $w_i$ ) and not IsContWord( $w_j$ )
(4c)    EW[i,j]+= $1 \times (1/m) \times srcWeight$ 
                elif IsContWord( $w_i$ ) and IsContWord( $w_j$ )
(4d)    EW[i,j]+= $1 \times m \times srcWeight$ 
(5) normalize each row of EW to sum to 1
//Iterate for PageRank
(6) set IP $1 \times v$  to
        [IntPrefs( $w_1$ ), IntPrefs( $w_2$ ), ..., IntPrefs( $w_v$ )]
(7) initialize IN $1 \times v$  to [ $1/v, 1/v, \dots, 1/v$ ]
    repeat
(8a) IN' =  $\lambda \times \mathbf{IN} \times \mathbf{EW} + (1 - \lambda) \times \mathbf{IP}$ 
(8b) normalize IN' to sum to 1
(8c) update IN with IN' after the check of IN and IN'
        until maxIter or avgDifference(IN, IN')  $\leq smallDiff$ 
(9) rankedInterests=Sort words in decreasing order of IN
    return the N rankedInterests with highest scores

```

Figure 1. Determining readers' words of interest.

Two semantic features are used in PageRank. Firstly, we weigh edges according to connecting words' syntactic parts-of-speech via edge multiplier *m*. We distinguish content words (e.g., nouns, verbs, adjectives and adverbs) from are not and implement three different levels of content-word score aggregation. Particularly, we have *slightly* content word centered score propagation when  $m > 1$  in Step (4a) and  $m = 1$  in Step (4b) to (4d), while we have *moderate* content word aggregation when  $m > 1$  in Step (4a) and (4d) and  $m = 1$  in Step (4b) and (4c). The third is to

*aggressively* make a non-content word's score flow to its content word partners by setting *m* in Step (4a) and  $1/m$  in Step (4b) where  $m > 1$ , and, circulate more  $w_i$ 's score to content words if  $w_i$  is a content word (i.e.,  $m > 1$  in Step (4c) and (4d)). The second semantic feature concerns source of words. Words may come from authors or readers, and *srcWeight* is set to  $\alpha$  if *st* is from *ART* and  $1 - \alpha$  otherwise. Smaller  $\alpha$ 's favor readers' perspectives more while functioning as a PageRank key-word extraction system if  $\alpha$  is one.

We set the one-by-*v* matrix **IP** of interest preference model using interest preferences for words in Step (6) and initialize the matrix **IN** of PageRank scores. Here we use word interestingness scores in Step (7). Then we re-distribute words' interestingness scores until the number of iterations or the average score differences of two consecutive iterations reach their respective limits. In each iteration, a word's interestingness score is the linear combination of its interest preference score and the sum of the propagation of its inbound words' previous PageRank scores. For the word  $w_j$  and any edge ( $w_i, w_j$ ) in *ART* and any edge ( $w_k, w_j$ ) in *qualityFB*, its new PageRank score is computed as

$$\mathbf{IN}'[1,j] = \lambda \times \left( \alpha \times \sum_{i \in v} \mathbf{IN}[1,i] \times \mathbf{EW}[i,j] + (1 - \alpha) \times \sum_{k \in v} \mathbf{IN}[1,k] \times \mathbf{EW}[k,j] \right) + (1 - \lambda) \times \mathbf{IP}[1,j]$$

Once the iterative process stops, we rank words according to their final interestingness scores and return *N* top-ranked words.

## 4 Experiments

In this section, we first present the data sets for training and evaluating *InterestFinder* (Section 4.1). Then, Section 4.2 reports the experimental results under different window sizes, content-word aggregation levels, estimation strategies of interest preferences.

### 4.1 Data Sets

We collected 6,600 articles from the blog website Wretch (www.wretch.cc) in November, 2012. In total, there were twelve first-level topics and 45 categories at the second tier. The example pre-defined two- to three-tier topic ontology ranged from Travel:Domestic to Life:Pets or from Fashion:Makeup to Technology:Games. Author-specified topic information was exploited to derive the estimation scores of interest preferences in Section 3.2. We also collected readers' feedback to the articles. We randomly chose 30 articles from training set for testing. Two human

judges annotated interested words after reading the articles in the test set.

	nDCG	P	MRR
<i>w/o</i>	.778	.397	.728
<i>agr@m=2</i>	.765	.390	.719
<i>mod@m=2</i>	.782	.390	.747
<i>slg@m=2</i>	<b>.792</b>	<b>.397</b>	.741

Table 1. System performance of different content-word aggregation levels at  $N=5$ .

## 4.2 Experimental Results

Our evaluation metrics are normalized discounted cumulative gain nDCG (Jarvelin and Kekalainen, 2002), precision (i.e., P), and mean reciprocal rank (i.e., MRR). We first examine the effectiveness of our semantic feature regarding content words in interest predictions. Table 1 suggests that while *slight* (*slg*) content word propagation is helpful, *moderate* (*mod*) and *aggressive* (*agr*) are not. Inflating content words’ statistics is simply sufficient. In addition, we found that smaller window size ( $WS=3$ ) fit more to our context of mixed-code blogs, while suitable window sizes were much larger in news articles and research abstracts (Liu et al., 2010).

Table 2 summarizes the interest prediction quality of two baselines, *entropy* and *tfidf*, and PageRank (PR) with different interest preference estimations on test set. In Table 2, *entropy* and *tfidf*, taking local (the article) and global (whole article collection) information into account, outperform PageRank using solely local information ( $PR+tf$ ). Among all,  $PR+tfidf$  achieves the best performance. Compared to  $PR+Pr$ ’s, *entropy* in  $PR+PrEntropy$ ’s does help to discern topical interest words. Moreover, the benefit of *entropy* is more evident when better estimation strategy  $Pr(tp|w)$  is applied: common words receive too much attention in  $Pr(w|tp)$  making readers’ interest words harder to come by.

(a) @ $N=5$	nDCG	P	MRR
<i>Entropy</i>	.677	.287	.659
<i>Tfidf</i>	.719	.313	.676
$PR+tf$	.657	.310	.632
$PR+Pr(w tp)$	.631	.290	.583
$PR+Pr(tp w)$	.673	.317	.639
$PR+PrEntropy(w tp)$	.636	.283	.584
$PR+PrEntropy(tp w)$	<b>.773</b>	<b>.337</b>	<b>.725</b>
$PR+tfidf$	<b>.792</b>	<b>.397</b>	<b>.741</b>

Table 2. System performance using article information alone at  $N=5$ .

We further exploit the collected reader feedback to train the baseline *tfidf* and our best sys-

tem  $PR+tfidf$ . Table 3 compares their interest predictions against judges’ interest and annotated words, within reader feedback, of interest in the articles. Note that the *tfidf* on reader feedback alone does not perform better.

(a) @ $N=5$	judges’ interest	general readers’ interest		
	nDCG	hit	nDCG	MRR
$(tfidf)_{none}$	.719	.10	.087	.075
$(tfidf)_{all}$	.699	.10	.079	.072
$(PR+tfidf)_{none}$	.792	.19	.137	.122
$(PR+tfidf)_{Coverage}$	.805	<b>.30</b>	<b>.186</b>	<b>.166</b>
$(PR+tfidf)_{Focus}$	.779	.27	.156	.137
$(PR+tfidf)_{Coverage+Focus}$	.794	<b>.30</b>	<b>.182</b>	<b>.164</b>

Table 3. System performance using *slg* at  $m=4$ ,  $WS=3$ ,  $\alpha=0.4$  and  $N=5$

In Table 3 we observe that (1) using all reader feedback is no better than using none (rows of *tfidf*) because not all feedback respond to the articles; (2) semantic feature of content source works well with *Coverage-* and *Focus-evaluate*. And *Coverage-* and *Focus-evaluate* are effective in checking informativity of social interaction data.  $(PR+tfidf)_{Coverage}$  or  $(PR+tfidf)_{Focus}$  achieves better performance on general readers’ interest while maintaining the prediction power on judges’ interest. (3) the chain of *Coverage-* and *Focus-evaluate*  $(PR+tfidf)_{Coverage+Focus}$  further prunes 6 and 12 percent of the reader sentences compared to the individual, and, encouragingly, using one-fourth of reader interactions still helps.

Based on the findings in Table 2 and 3, we believe that proposed interest preference models, semantic features (i.e, content source and content word), and the informativity check on social interaction content are simple yet helpful in suggesting good and representative reader interests.

## 5 Conclusion

We have introduced a method for predicting reader interest in an article. In interest prediction, we turn to social interaction content instead of reader profile and browse history. The method involves estimating topical interest preferences, screening public reader responses, and leveraging semantic features such as words’ sources (i.e., from article authors or readers) and words’ parts-of-speech in PageRank. We have implemented and evaluated the method as applied to interest analysis. In two separate evaluations, we have shown that *quality* social interaction content and semantic aware PageRank help to accurately cover broader spectrum of reader interest.



## Acknowledgements

Research of this paper was partially supported by National Science Council, Taiwan, under the contract NSC101-2628-E-224-001-MY3.

## References

- Scott A. Golder and Bernardo A. Huberman. 2006. Usage patterns of collaborative tagging systems. *Information Science*, 32(2): 198-208.
- Harry Halpin, Valentin Robu, and Hana Shepherd. 2007. The complex dynamics of collaborative tagging. In *Proceedings of the WWW*, pages 211-220.
- Kalervo Jarvelin and Jaana Kekalainen. 2002. Cumulated gain-based evaluation of IR technologies. *ACM Transactions on Information Systems*, 20(4): 422-446.
- Hongxia Jin. 2012. Content recommendation for attention management in unified social messaging. In *Proceedings of the AAAI*, pages 627-633.
- Quanzhi Li, Yi-Fang Wu, Razvan Bot, and Xin Chen. 2004. Incorporating document keyphrases in search results. In *Proceedings of the Americas Conference on Information Systems*.
- Zhenhui Li, Ging Zhou, Yun-Fang Juan, and Jiawei Han. 2010. Keyword extraction for social snippets. In *Proceedings of the WWW*, pages 1143-1144.
- Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the EMNLP*, pages 366-376.
- Chris D. Manning and Hinrich Schütze. 2000. *Foundations of statistical natural language processing*. MIT Press.
- Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing orders into texts. In *Proceedings of the EMNLP*, pages 404-411.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311-318.
- Manos Tsagkias and Roi Blanco. 2012. Language intent models for inferring user browsing behavior. In *Proceedings of the SIGIR*, pages 335-344.
- Ryen W. White, Peter Bailey, and Liwei Chen. 2009. Predicting user interest from contextual information. In *Proceedings of the SIGIR*, pages 363-370.
- Wei Wu, Bin Zhang, and Mari Ostendorf. 2010. Automatic generation of personalized annotation tags for Twitter users. In *Proceedings of the NAACL*, pages 689-692.
- Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation- a generative model approach. In *Proceedings of the SIGIR*, pages 671-680.
- Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyword extraction from Twitter. In *Proceedings of the ACL*, pages 379-388.

# Time Series Topic Modeling and Bursty Topic Detection of Correlated News and Twitter

**Daichi Koike**  
**Yusuke Takahashi**  
**Takehito Utsuro**  
Grad. Sch. Sys. & Inf. Eng.,  
University of Tsukuba,  
Tsukuba, 305-8573, JAPAN

**Masaharu Yoshioka**  
Grad. Sch. Inf. Sci. & Tech.,  
Hokkaido University,  
Sapporo, 060-0808,  
JAPAN

**Noriko Kando**  
National Institute  
of Informatics,  
Tokyo, 101-8430,  
JAPAN

## Abstract

News and twitter are sometimes closely correlated, while sometimes each of them has quite independent flow of information, due to the difference of the concerns of their information sources. In order to effectively capture the nature of those two text streams, it is very important to model both their correlation and their difference. This paper first models their correlation by applying a time series topic model to the document stream of the mixture of time series news and twitter. Next, we divide news streams and twitter into distinct two series of document streams, and then we apply our model of bursty topic detection based on the Kleinberg's burst detection model. This approach successfully models the difference of the two time series topic models of news and twitter as each having independent information source and its own concern.

## 1 Introduction

The background of this this paper is in two types of modeling of information flow in news stream, namely, burst analysis and topic modeling. Both types of modeling, to some extent, aim at aggregating information and reducing redundancy within the information flow in news stream.

First, when one wants to detect a kind of topics that are paid much more attention than usual, it is usually necessary for him/her to carefully watch every article in news stream at every moment. In such a situation, it is well known in the field of time series analysis that Kleinberg's modeling of bursts (Kleinberg, 2002) is quite effective in detecting burst of keywords. Second, topic models such as LDA (latent Dirichlet allocation) (Blei et

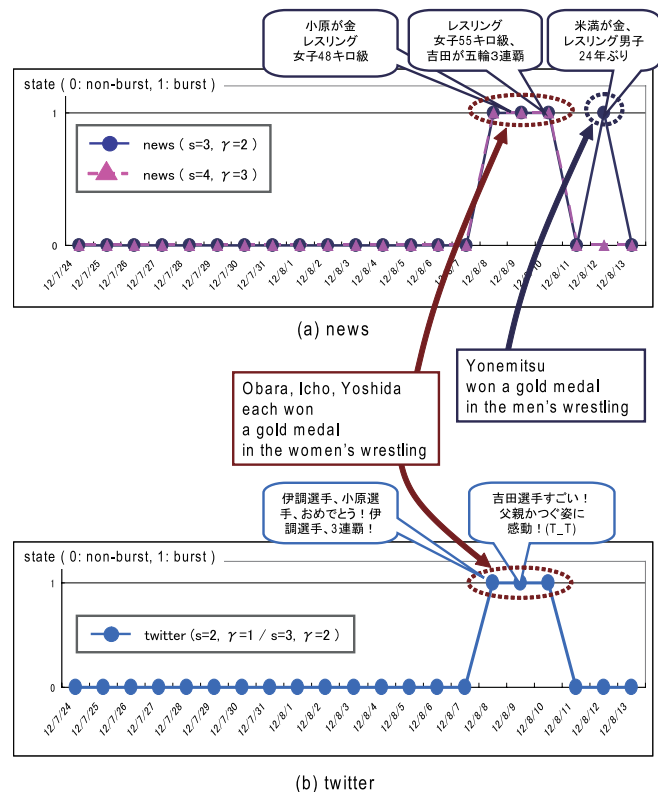


Figure 1: Optimal State Sequence for the Topic “wrestling”

al., 2003) and DTM (dynamic topic model) (Blei and Lafferty, 2006) are also quite effective in estimating distribution of topics over a document collection such as articles in news stream. Unlike LDA, in DTM, we suppose that the data is divided by time slice, for example by date. DTM models the documents (such as articles of news stream) of each slice with a  $K$ -component topic model, where the  $k$ -th topic at slice  $t$  smoothly evolves from the  $k$ -th topic at slice  $t - 1$ .

Based on those arguments above, Takahashi et al. (2012) proposed how to integrate the two types of modeling of information flow in news stream. Here, it is important to note that Kleinberg's modeling of bursts is usually applied only to bursts of keywords but not to those of topics. Thus, Taka-

hashi et al. (2012) proposed how to apply Kleinberg’s modeling of bursts to topics estimated by a topic model such as DTM. Typical results of applying the technique to time series news stream can be illustrated as in Figure 1 (a). In this example, we first estimate time series topics through DTM, among which is the one “wrestling” as shown in this figure. Then, we can detect the burst of the topic on the dates when those two Japanese wrestlers won the gold medals.

Unlike Takahashi et al. (2012), this paper studies the issue of time series topic modeling and bursty topic detection of possibly correlated news and twitter. News and twitter are sometimes closely correlated, while sometimes each of them has quite independent flow of information, due to the difference of the concerns of their information sources. In order to effectively capture the nature of those two text streams, it is very important to model both their correlation and their difference. This paper first models their correlation by applying a time series topic model to the document stream of the mixture of time series news and twitter. This approach successfully models the time series topic models of news and twitter as closely correlated to each other. Next, we divide news streams and twitter into distinct two series of document streams, and then we apply our model of bursty topic detection based on the Kleinberg’s burst detection model. With this procedure, we show that, even though we estimate the time series topic model with the document stream of the mixture of news and twitter, we can detect bursty topics individually both in the news stream and in twitter. This approach again successfully models the difference of the two time series topic models of news and twitter as each having independent information source and its own concern.

## 2 Time Series Documents Set for Evaluation

In this paper, we collect time series news articles of a certain period as well as tweets texts of the same period that are closely related to the news articles. Then, we construct a time series document set consisting of the mixture of the news articles and tweets texts (Table 1) and use it for evaluation.

### 2.1 News

As the news stream documents set for evaluation, during the period from July 24th to August 13th,

Table 1: Time Series Documents Set

news articles	tweets	total # of document
2,308	57,414	59,722

2012, we collected 3,157 Yomiuri newspaper articles, 4,587 Nikkei newspaper articles, and 3,458 Asahi newspaper articles which amount to 11,202 newspaper articles in total<sup>1</sup>. Then, we select a subset of the whole 11,202 newspaper articles which are related to “the London Olympic game”, where we collect 2,308 articles that contain at least one of 8 keywords<sup>2</sup> into the subset. The subset consists of 659 Yomiuri newspaper articles, 679 Nikkei newspaper articles, and 970 Asahi newspaper articles.

### 2.2 Twitter

As the tweet text data set for evaluation, during the period from July 24th to August 13th, 2012, we collected 9,509,774 tweets from the Twitter<sup>3</sup> with the Streaming API. Then, we removed tweets with official retweets and those including URLs, and 7,752,129 tweets remained. Finally, we select a subset which are related to “the London Olympic game”. Here, we collect 57,414 tweets that contain at least one of the 8 keywords listed above, which are closely related to “the London Olympic game”, into the subset.

## 3 Kleinberg’s Bursts Modeling

Kleinberg (2002) proposed two types of frameworks for modeling bursts. The first type of modeling is based on considering a sequence of message arrival times, where a sequence of messages is regarded as bursty if their inter-arrival gaps are too small than usual. The second type of modeling is, on the other hand, based on the case where documents arrive in discrete *batches* and in each batch of documents, some are *relevant* (e.g., news text contains a particular word) and some are *irrelevant*. In this second type of bursts modeling, a sequence of batched arrivals could be considered bursty if the fraction of relevant documents alternates between reasonably long periods in which the fraction is small and other periods in which it is large. Out of the two modelings, this paper em-

<sup>1</sup><http://www.yomiuri.co.jp/>, <http://www.nikkei.com/>, and <http://www.asahi.com/>.

<sup>2</sup>五輪 (*Gorin* (“Olympic” in Chinese characters)), ロンドン (London), オリンピック (Olympic (in katakana characters)), 金メダル (gold medal), 銀メダル (silver medal), 銅メダル (bronze medal), 選手 (athlete), 日本代表 (Japanese national team)

<sup>3</sup><https://twitter.com/>

employs the latter, which is named as *enumerating bursts* in Kleinberg (Kleinberg, 2002).

#### 4 Applying Time Series Topic Model

As a time series topic model, this paper employs DTM (dynamic topic model) (Blei and Lafferty, 2006). In this paper, in order to model time series news stream in terms of a time series topic model, we consider date as the time slice  $t$ . Given the number of topics  $K$  as well as time series sequence of batches each of which consists of documents represented by a sequence of words  $w$ , on each date  $t$  (i.e., time slice  $t$ ), DTM estimated the distribution  $p(w|z_n)$  ( $w \in V$ , the vocabulary set) of a word  $w$  given a topic  $z_n$  ( $n = 1, \dots, K$ ) as well as that  $p(z_n|b)$  ( $n = 1, \dots, K$ ) of a topic  $z_n$  given a document  $b$ , where  $V$  is the set of words appearing in the whole document set. In this paper, we estimate the distributions  $p(w|z_n)$  ( $w \in V$ ) and  $p(z_n|b)$  ( $n = 1, \dots, K$ ) by a Blei’s toolkit<sup>4</sup>, where the parameters are tuned through a preliminary evaluation as the number of topics  $K = 50$  as well as  $\alpha = 0.01$ . The DTM topic modeling toolkit is applied to the time series document set shown in Table 1, which consists of the mixture of the news articles and tweets texts. Here, as a word  $w$  ( $w \in V$ ) constituting each document, we extract Japanese Wikipedia<sup>5</sup> entry titles as well as their redirects.

#### 5 Modeling Bursty Topics Independently from News and Twitter

In this section, we are given a time series document set which consists of the mixture of two types of documents originating from two distinct sources, e.g., news and tweets. In this situation, we assume that a time series topic model is estimated with the mixture of two types of time series documents, where the distinction of the two sources is ignored at the step of time series topic model estimation. Then, the following procedure presents how to model bursty topics for each of the two types of time series documents independently. This means, in the case of news and twitter, that, although the time series topic model is estimated with the mixture of time series news articles and tweets texts, bursty topics are detected independently from news and twitter.

<sup>4</sup><http://www.cs.princeton.edu/~blei/topicmodeling.html>

<sup>5</sup><http://ja.wikipedia.org/>

In this bursty topic modeling, first, we suppose that, on the date  $t$  (i.e., time slice  $t$ ), we have two types of documents  $b_x$  and  $b_y$  each of which originates from the source  $x$  and  $y$ , respectively. Then, for the source  $x$ , we regard a document  $b_x$  as *relevant* to a certain topic  $z_n$  that are estimated through the DTM topic modeling procedure, to the degree of the amount of the probability  $p(z_n|b_x)$ . Similarly for the source  $y$ , we regard a document  $b_y$  as *relevant* to a certain topic  $z_n$ , to the degree of the amount of the probability  $p(z_n|b_y)$ . Next, for the source  $x$ , we estimate the number  $r_{t,x}$  of relevant documents out of a total of  $d_{t,x}$  simply by summing up the probability  $p(z_n|b_x)$  over the whole document set (similarly for the source  $y$ ):

$$r_{t,x} = \sum_{b_x} p(z_n|b_x) \quad r_{t,y} = \sum_{b_y} p(z_n|b_y)$$

Once we have the number  $r_{t,x}$  and  $r_{t,y}$  for the sources  $x$  and  $y$ , then we can estimate the total number of relevant documents throughout the whole batch sequence  $\mathbf{B} = (B_1, \dots, B_m)$  as  $R_x = \sum_{t=1}^m r_{t,x}$  and  $R_y = \sum_{t=1}^m r_{t,y}$ . Denoting the

total numbers of documents on the date  $t$  for the sources  $x$  and  $y$  as  $d_{t,x}$  and  $d_{t,y}$ , respectively, we have the total numbers of documents throughout the whole batch sequence as  $D_x = \sum_{t=1}^m d_{t,x}$  and

$D_y = \sum_{t=1}^m d_{t,y}$ , respectively. Finally, we can estimate

the expected fraction of relevant documents as  $p_{0,x} = R_x/D_x$  and  $p_{0,y} = R_y/D_y$ , respectively. Then, by simply following the formalization of bursty topics we proposed in Takahashi et al. (2012), it is quite straightforward to model bursty topics independently for each of the two sources  $x$  and  $y$ . In the following evaluation, we consider the sources  $x$  and  $y$  as time series news articles and tweet texts shown in Table 1. As the two parameters  $s$  and  $\gamma$  for bursty topic detection<sup>6</sup>, we compare two pairs  $s = 4, \gamma = 3$  and  $s = 3, \gamma = 2$  for time series news articles, and  $s = 3, \gamma = 2$  and  $s = 2, \gamma = 1$  for tweets text.

## 6 Evaluation

### 6.1 The Procedure

As the evaluation of the proposed technique, we examine the correctness of the detected bursty top-

<sup>6</sup> $s$  is a parameter for scaling expected fractions of relevant documents between burst / non-burst states.  $\gamma$  is a parameter for the cost of moving from the non-burst state to the burst state. The details of the two parameters are described in Kleinberg (2002) and Takahashi et al. (2012).

Table 2: Evaluation Results: Precision of Detecting Bursty Topics (for 34 Topics relevant to “the London Olympic Games” out of the whole 50)

	bursts detected in both news and twitter	bursts detected only in one of news and twitter
news	per day: 87.5 % (14/16)	per day: 100 % (2/2), per topic: 100 % (1/1)
twitter	per topic 87.5 % (7/8)	per day: 100 % (32/32), per topic: 100 % (13/13)

ics. For each topic  $z_n$ , collect the documents  $b$  which satisfies  $z_n = \operatorname{argmax}_{z'} p(z'|b)$  into the set  $B_{1st}(z_n)$ . Then, we first judge whether most of the collected documents (both news articles and tweets texts)  $b \in B_{1st}(z_n)$  have relatively similar contents. If so, next we examine the correctness of the detected burst of that topic.

We evaluate the detected bursty topics per day or per topic. As for “per day evaluation”, we examine whether, on each day of the burst, the detected burst is appropriate or not. As for “per topic evaluation”, we examine whether, for each topic, all of the detected bursts are appropriate or not.

Out of the whole 50 topics, we manually select 34 that are relevant to “the London Olympic games”, and show the evaluation results of detecting bursty topics in Table 2. Here, as the two parameters  $s$  and  $\gamma$  for bursty topic detection, we show those with  $s = 4$  and  $\gamma = 3$  for news and  $s = 3$  and  $\gamma = 2$  for tweets, for which we have the highest precision in bursty topic detection. We also classify the detected bursts per day and detected bursty topics (i.e., *per topic*) into the following two types: (a) *the bursty topic is shared between news and twitter*, and (b) *the bursty topic is detected only in one of news and twitter*.

## 6.2 Evaluation Results

As shown in Table 2, for the bursty topic of type (b), precisions for both “per day” and “per topic” evaluation are 100% (both for news and twitter). The proposed technique is quite effective in detecting many bursty topics that are observed only in twitter. For the bursty topic of type (a), over detection of bursty topics is only for one topic, which is about “*politics*”. The reason why this over detection occurred is mainly because we observed fewer numbers of news articles and tweets on politics during the period of “the London Olympic games”, and then, the periods other than “the London Olympic games” are detected as bursty. Also

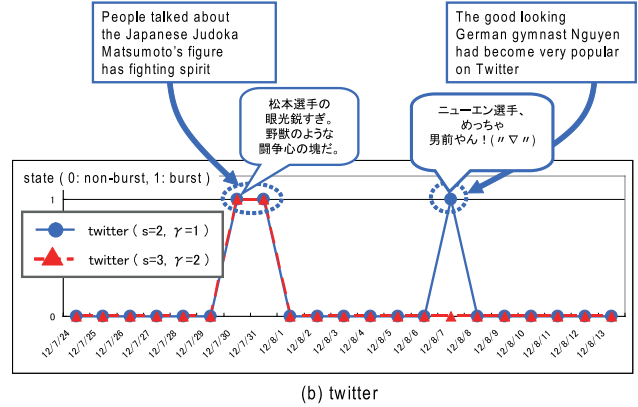


Figure 2: Optimal State Sequence for the Topic “good looking athletes” (observed only in twitter)

for the bursty topic of type (a), reasons of bursts in news articles and tweets texts are almost the same as each other. This result clearly supports our claim that the proposed technique is quite effective in detecting closely related bursty topics in news and twitter.

Figure 1 plots the optimal state sequence for the topic “wrestling” for both news and twitter. For this topic, some of the bursts are shared between news and twitter, so we also show the results of aligning bursts between news and twitter. Figure 2 also plots the optimal state sequence for the topic “good looking athletes”, where for this topic, all the documents are from the source twitter and bursts are detected only for twitter<sup>7</sup>.

## 7 Conclusion

This paper showed that, even though we estimate the time series topic model with the document stream of the mixture of news and twitter, we can detect bursty topics independently both in the news stream and in twitter. Among several related works, Diao et al. (2012) proposed a topic model for detecting bursty topics from microblogs. Compared with Diao et al. (2012), one of our major contributions is that we mainly focus on the modeling of correlation and difference between news and twitter.

<sup>7</sup>It is surprising that tweets that mentioned good looking athletes are collected altogether in this topic. Many tweets collected in this topic on the non-bursty days said that he/she likes a certain athlete. And, those tweets share the terms 選手 (athlete) and 好き (like). But, especially on the days when the bursts were observed, much more people posted that the Japanese judoka Matsumoto and the German gymnast Nguyen were so impressive because of their looking. This is why we observed bursts on those days.

## References

- D. M. Blei and J. D. Lafferty. 2006. Dynamic topic models. In *Proc. 23rd ICML*, pages 113–120.
- D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim. 2012. Finding bursty topics from microblogs. In *Proc. 50th ACL*, pages 536–544.
- J. Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proc. 8th SIGKDD*, pages 91–101.
- Y. Takahashi, T. Utsuro, M. Yoshioka, N. Kando, T. Fukuhara, H. Nakagawa, and Y. Kiyota. 2012. Applying a burst model to detect bursty topics in a topic model. In *JapTAL 2012*, volume 7614 of *LNCS*, pages 239–249. Springer.

# A Distant Supervision Approach for Identifying Perspectives in Unstructured User-Generated Text

**Attapol Thamrongrattanarit\***

Brandeis University  
Waltham, MA  
tet@brandeis.edu

**Benjamin Goldenberg**

Yelp Inc  
San Francisco, CA  
benjamin@yelp.com

**Colin Pollock**

Yelp Inc  
San Francisco, CA  
cpollock@yelp.com

**Jason Fennell**

Yelp Inc  
San Francisco, CA  
jfennell@yelp.com

## Abstract

With the overabundance of online user-generated content, the ability to filter based on relevant perspectives is becoming increasingly important. Identifying the perspective of the authors with the review text would enhance the retrieval of pertinent information. This problem can be traditionally formulated as a text classification task and solved by annotating the data and building a supervised learning system. However, rare classes might render annotation even more difficult and expensive. Here, we used a distant supervision approach to identify restaurant reviews that were written from the perspective of a vegetarian, and we achieved a macro-average F1 score of 79.40% with minimal annotation effort.

## 1 Introduction

The center of the information world has shifted from select few authorities to the global wisdom of the crowd and user-generated content. While useful and large, the volume of the information requires efficient organization, information retrieval, and data mining techniques to select the most relevant contents. For example, restaurant goers looking for vegetarian-friendly restaurants might want to read restaurant reviews that are written by a vegetarian. Authors' perspectives potentially provide a meaningful axis along which the documents can be organized.

Past studies have formulated this problem as a document-level supervised text classification

---

\*The author conducted the work during his internship at Yelp.

problem (Manning et al., 2008), but the supervised learning paradigm might not be suitable in certain scenarios. Supervised learning algorithms can achieve superior performance compared to unsupervised learning algorithms at the expense of costly annotation efforts in creating labeled datasets for the algorithms to learn from. The perspectives that we would like to identify, however, might be very specific and somewhat rare in the document collection. To continue with the restaurant review example, only an estimated of 3.2% of the U.S. population identify themselves as vegetarian (Haddad and Tanzman, 2003), so the restaurant reviews written from the perspective of a vegetarian might be very rare in the corpus.

This rare class problem necessitates a larger number of annotated documents to collect sufficient positive examples. For instance, approximately 10,000 data instances must be labeled in order to obtain a mere 320 data points for vegetarian reviews. Additionally, the resulting classifier trained on a specific annotated corpus will tend to be biased toward that text domain, and the performance might degenerate when the classifier is applied to text in another domain. Although the rare class problem is well studied in supervised settings (He and Ghodsi, 2010; Joshi et al., 2001), to our knowledge we have not encountered a distantly supervised algorithm applied to a dataset where a rare class is of interest.

Distant supervision approaches address these problems by exploiting prior knowledge or external resources to gather large number of training data or features to train a classifier without manual annotation. A distant supervision algorithm might start by using a set of simple rules or a knowledge base to form distant supervision criteria (Mintz et al., 2009), then create an ini-

tial training set from such criteria. The labels of these initial training samples are sometimes said to be *weakly labeled* because they are not individually supervised or manually labeled by a human annotator. Instead, the labels are distantly supervised by heuristics or informed by an extensive knowledge base. For instance, a distant supervision approach has been applied to Twitter data, which is massive and hard to annotate (Marchetti-Bowick and Chambers, 2012). Emoticons were used to identify tweets with positive or negative sentiments, which then served as training samples for supervised classifiers (Go et al., 2009). Notably, distant supervision approaches were successfully used in relation extraction. Mintz et al. (2009) employed a knowledge base to extract patterns and features for a relation extraction system, where the supervised training data were relatively small and domain specific.

The supervised learning paradigm might not suit ever-growing user-generated datasets that contain rare classes and suffer from prohibitive annotation cost. Here, we present a distant supervision approach for identifying rare author’s perspectives in unstructured user-generated content. This method alleviates the problem of rare classes and reduces the time and cost of annotating data.

## 2 Corpus and Task Description

We randomly selected ten million user-generated restaurant reviews in English from yelp.com, a consumer review website.<sup>1</sup> The review authors’ personal information was removed from the data. Most of the text consists of well formed sentences, due to the greater character limit than, say, Twitter. Although, like most unstructured user-generated corpora, these reviews contain typos and non-standard structure, e.g. ASCII art and use of dashes as bullet points. Each review contains 151.04 tokens on average, so we have a total of approximately 1.5 billion tokens in this dataset.

For this corpus, we focus on identifying the restaurant reviews that are written from the perspective of a vegetarian. We annotated a small number of reviews to use as a test set for final evaluation. Each review was annotated independently by two annotators. The inter-annotator agreement is moderate (Cohen’s  $\kappa = 0.58$ ). Only 34 reviews out of 1,021 labeled reviews are labeled as written

---

<sup>1</sup>The corpus is available upon request on the website [www.yelp.com/academic\\_dataset](http://www.yelp.com/academic_dataset)

from the perspective of a vegetarian, which suggests that this perspective occurs rarely in the corpus.

## 3 Methodology

We employed simple phrase matching to collect weakly labeled data. If a review contains the phrase “I’m a vegetarian,” “I’m vegetarian,” or “As a vegetarian, I,” we regard those reviews as written from the perspective of a vegetarian. On the other hand, if a review mentions beef, pork, or chicken without the word “fake” preceding it, then such review is tagged as not written from a perspective of a vegetarian. These simple phrase matching rules are applied to every review in the corpus. The reviews that are not selected by the rules are discarded from the weakly labeled training data.

This weakly labeled training set is used to train a two-way classification Multinomial Naive Bayes model. The classifier uses all of the words in the documents as features. Weakly labeled data are noisier than manually labeled data, which might cause the classifier to be less generalizable due to overwhelming noisy features. Thus, we perform feature selection by using the Bayesian Information Criterion (BIC) to reduce noise and improve the performance (Schwarz, 1978; Forman, 2003). We noted that the proportion of each perspective in the weakly labeled data does not necessarily match the true proportion. We therefore manually set the prior probabilities of the labels to 0.9 and 0.1 for non-vegetarian and vegetarian respectively.

In general, one can use any arbitrary criteria to cull weakly labeled data from the corpus, as long as the criteria are high in precision. If the corpus is massive, which is usually the case in user-generated content on the web, then we afford to sacrifice recall for less noisy training instances. With regard to training classifiers, one can choose any supervised learning algorithm.

## 4 Experiment Setup and Results

Our distant supervision criteria identified 12,514 reviews written from the perspective of a vegetarian, and 3,076,256 reviews not written from such perspective. 7,193,878 reviews were left unannotated and discarded because they don’t contain any of the phrases in our criteria. These reviews constitute weakly labeled training data and account for roughly 30% of the original unlabeled data of ten million reviews.



	Precision	Recall	$F_1$
Vegetarian	80.85	80.07	80.46
Non-vegetarian	97.50	97.62	97.56
Macro-average	89.17	88.84	89.01

Table 1: 7-fold cross-validation results based on the weakly labeled data. The classifier achieved the macro-averaged  $F_1$  measure of 89.01.

#### 4.1 Experiment 1

To evaluate the distant supervision method as it is applied to this task, we ran a 7-fold cross-validation on the weakly labeled training data and computed precision, recall, and  $F_1$  measure for each class. The words used to collect weakly labeled data build the nearly perfect classifier features, therefore we excluded those words from the feature set before training the classifier. The results are shown in Table 1. We achieve the macro-average  $F_1$  score of 89.01% and the accuracy rate of 95.67%.

#### 4.2 Experiment 2

The evaluation based on the 7-fold cross-validation over the weakly labeled data might not accurately reflect how well the resulting classification will perform when applied to the unlabeled dataset. We evaluated the classifier on the manually annotated test set detailed in the earlier section. Like the first experiment, all of the words and phrases involved in gathering weakly labeled data were excluded from the feature set for training a classifier. The test set has a total of 1,021 labeled reviews, none of which overlaps with the original unlabeled dataset.

The classifier was evaluated on four different subsets of the test set to see the performance of the system in different scenarios:

1. All reviews in the test set. The reviews are for restaurants which also include bars and coffee shops, where the perspectives of vegetarians are even more rare or not applicable.
2. All reviews in the test set that are longer than 250 characters. Some reviews are too short to contain useful information for the vegetarian perspectives. This subset contains 623 reviews.
3. Food reviews only. In this scenario, we exclude reviews for bars and coffee shops. We

were left with 316 reviews.

4. Food reviews that are longer than 250 characters. This subset contains 270 reviews.

The classifier attained the best performance when tested on food reviews longer than 250 characters. In this scenario, it achieved the macro-averaged  $F_1$  score of 79.40% and an accuracy rate of 92.22%. The baseline accuracy by guessing non-vegetarian for all reviews is 88.88%. The performance report for all scenarios is summarized in Table 2.

The  $F_1$  scores for vegetarian perspectives are lower across all experimental conditions possibly because of the highly imbalanced label distribution or insufficient positive training samples. Since any supervised algorithms can be integrated with our distant supervision approach, these problems can be remedied by downsampling or models that are robust to imbalanced data (Japkowicz, 2000).

It is important to note that the set of rules alone do not make any prediction on the label of our test set, because the phrases that constitute the rules do not match any of the reviews in the test set. Therefore, the distantly supervised training is necessary to build a classifier.

## 5 Discussion

In our proposed method for identifying vegetarian-written reviews, we exploited the fact that some of the reviews are already weakly labeled, motivating the words and phrases used for the distant supervision criteria. The key step for this approach is writing rules or criteria for collecting the weakly labeled data. As we focus on massive unstructured user-generated text, the criteria we use must be highly precise to prevent mislabeled data from being introduced into the training process. Although high-precision rules by definition will only recognize a small percentage of the positive samples, the problem is remedied by the fact that user-generated data are massive and constantly grow larger. In this study, our restrictive rules yield 12,514 reviews written from the perspective of a vegetarian, which is approximately 0.001% of the original dataset but suffices to build a classifier, as shown by the evaluation result.

Our approach can be thought of as similar to the bootstrapping technique, which has been explored extensively in the context of relation extraction (Gabbard et al., 2011) and text classifica-

	size	Vegetarian			Non-vegetarian			Overall	
		P	R	F1	P	R	F1	Acc.	F1
Long food reviews	270	66.67	60.00	63.15	95.06	96.25	95.65	92.22	79.40
Long reviews only	623	71.42	50.00	58.82	93.97	97.50	95.70	92.22	77.26
Food reviews only	316	64.51	58.82	61.53	95.08	96.09	95.59	92.08	78.56
All reviews	1,021	32.22	58.52	41.66	98.54	95.74	97.12	94.51	69.39

Table 2: Evaluation result based on the manually annotated test set. The reviews that contain more than 250 characters are considered long. The system performs the best when tested with long food reviews only.

tion (Mccallum, 1999). A bootstrapping algorithm starts with a small set of annotated seed training instances. Classifier training or pattern extraction is done based on the seed instances and then used to reap more training instances from the unlabeled data. This cycle continues for multiple iterations, and the performance is monitored at each iteration to ensure the improvement. A downside of this approach is that one bad iteration might introduce many mislabeled instances, degenerating the algorithm. In practice, extra human supervision must then check if the new training instances in each iteration are acceptable (Freedman et al., 2011). Our distant supervision approach requires human supervision only when writing criteria for initial training instances.

Unlike bootstrapping, our approach trains the classifier only once. Therefore, classifiers that take long to train such as Support Vector Machine can be trained within reasonable amount of time. When paired with automatic feature selection like the one used in this study, building a distantly supervised classifier is a matter of coming up with high-precision criteria to initiate the algorithm. If the criteria for distant supervision are precise enough, very little noise will be introduced into the training instances. These characteristics of our approach are attractive for massive data from user-generated content that might render computational cost of bootstrapping too costly.

## 6 Conclusion and Future Directions

We presented a distant supervision approach to identify authors’ perspectives in an unstructured, user-generated, and possibly massive corpus. Our experiment shows that high-precision restrictive rules can potentially gather weakly labeled data to train a classifier robust enough to perform well on the rest of the corpus. This method demonstrates the potential to enhance user experi-

ence on a user-generated business review website like www.yelp.com, by allowing an information-retrieval system that can fetch documents based on authors’ perspectives.

As a future direction, this similar method can be applied to massive user-generated microblogs like Twitter data to identify authors’ perspectives. For example, if one could identify each tweet as written by a Republican or a Democrat, one might be able to mine opinions from each political party separately. One could also endeavor to identify documents written by the same authors. For instance, a vegetarian is more likely to write from the perspective of a vegetarian, so we can restrict distant supervision rules to require consistent labels for the same authors in order for the reviews to enter the training set.

## References

- George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305.
- Marjorie Freedman, Lance Ramshaw, Elizabeth Boschee, Ryan Gabbard, Gary Kratkiewicz, Nicolas Ward, and Ralph Weischedel. 2011. Extreme extraction: machine reading in a week. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1437–1446. Association for Computational Linguistics.
- Ryan Gabbard, Marjorie Freedman, and Ralph Weischedel. 2011. Coreference for learning to extract relations: yes, virginia, coreference matters. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2, pages 288–293.
- Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12.

- Ella H Haddad and Jay S Tanzman. 2003. What do vegetarians in the united states eat? *The American journal of clinical nutrition*, 78(3):626S–632S.
- He He and Ali Ghodsi. 2010. Rare class classification by support vector machine. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 548–551. IEEE.
- Nathalie Japkowicz. 2000. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, volume 68.
- Mahesh V Joshi, Ramesh C Agarwal, and Vipin Kumar. 2001. Mining needle in a haystack: classifying rare classes via two-phase rule induction. In *ACM SIGMOD Record*, volume 30, pages 91–102. ACM.
- Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.
- Micol Marchetti-Bowick and Nathanael Chambers. 2012. Learning for microblogs with distant supervision: Political forecasting with twitter. *EACL 2012*, page 603.
- Andrew McCallum. 1999. Text classification by bootstrapping with keywords, em and shrinkage. In *In ACL99-Workshop for Unsupervised Learning in Natural Language Processing*.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.
- Gideon Schwarz. 1978. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

# An Approach of Hybrid Hierarchical Structure for Word Similarity Computing by HowNet

**Jiangming Liu**  
Beijing Jiaotong University  
jmlunlp@gmail.com

**Jinan Xu**  
Beijing Jiaotong University  
jaxu@bjtu.edu.cn

**Yujie Zhang**  
Beijing Jiaotong University  
yjzhang@bjtu.edu.cn

## Abstract

Word similarity computing is an important and fundamental task in the field of natural language processing. Most of word similarity methods perform well in synonyms, but not well between words whose similarity is vague. To overcome this problem, this paper proposes an approach of hybrid hierarchical structure computing Chinese word similarity to achieve fine-grained similarity results with HowNet 2008. The experimental results prove that the method has a better effect on computing similarity of synonyms and antonyms including nouns, verbs and adjectives. Besides, it performs stably on standard data provided by SemEval 2012.

## 1 Introduction

Word similarity computing plays an important role in various fields, such as Natural Language Understanding and Cognitive Science (Bunescu and Huang, 2010b; Mohler et al., 2011; Wang and Wan, 2011;). Moreover, it is a pivotal method in Word Sense Disambiguation (WSD).

Two main types of word similarity computing methods have been proposed. One is usually based on the thesaurus. The methods of this type utilize the structure of thesaurus (Liu and Li, 2002; Ge et al., 2010) with the advantages of preciseness and deep usage of word semantics, but a relatively complete semantic dictionary is required in order to ensure the presence of words in thesaurus. The other methods are based on large-scale corpora with some inevitable disadvantages, such as the frequent need of large-scale corpora, noise, low search efficiency etc. (Nakov and Hearst, 2008). Therefore, it is fine to create a refined thesaurus with Internet resource or large-scale corpora (Morita et al., 2011; Navigli and Ponzetto, 2010; Davidov and Rappoport, 2010) as an interim for computing word similarity.

WordNet is deemed to be very valuable thesaurus. Since Chinese that belongs to isolated

language is different from English that belongs to inflected language and the complex Chinese grammar is highly ambiguous, computing Chinese words similarity is more difficult than English under the same lack of systematic resource. HowNet is also a valuable bilingual knowledge thesaurus organized by Zhongdong Dong.

HowNet uses a markup language called KDML to describe word's concept which facilitates computer processing (Li et al., 2012). A different semantic of one word has a different DEF description. DEF is defined by a number of sememes and the descriptions of semantic relations between words. It is worth to mention that sememes are the most basic and the smallest units which cannot be easily divided (Liu and Li, 2002), and they are extracted from about six thousands of Chinese characters (Dong and Dong, 2006). An example of one DEF of *saleslady* can be described as a tree-like structure (Figure 1). The details of description in HowNet can be accessed in the paper (Dong, 2002).

In closely related works, Liu (2002) proposed an up-down algorithm on HowNet 2000 and achieved a good result. Li (2012) proposed an algorithm based on the hierarchic DEF description of words on HowNet 2002. In HowNet 2008, hierarchic DEF (Dong and Dong, 2002) definition is involved not only in words, but also in sememes. The algorithm proposed by Liu is useful, especially in example-based machine translation. The algorithm proposed by Li is detailedly experimented only in synonyms. The algorithm proposed in this paper fuses hierarchic DEF definition of sememe and hierarchic structure of sememe. It performs better and more stably both between the high similarity words namely synonyms and between the vague similarity words.

The remainder of the paper is organized as follows: Section 2 describes our algorithm in detail. Section 3 presents the experimental results and comparison. In the last section, conclusions are put forward and future work is discussed.

## 2 Similarity Computing

### 2.1 DEF similarity computing

The hierarchy of DEF is introduced as a tree-like structure. Due to different relation on the edge of trees, computing DEF similarity, unlike conventional tree similarity is one of our core works.

The similarity between one pair of nodes in the same layer of tree comes from two types of similarity, namely the relation similarity from that of its children nodes and sememe similarity itself which is described later in detail in section 2.2.

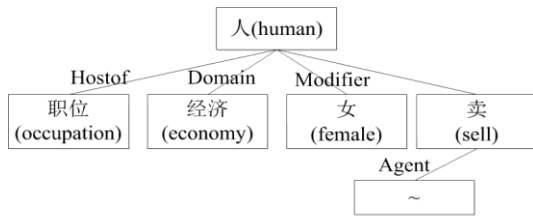


Figure 1. DEF hierarchy of *saleslady*

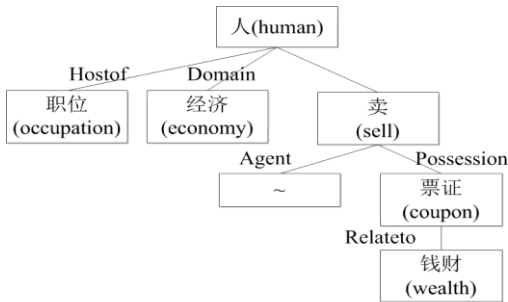


Figure 2. DEF hierarchy of *conductor*

For relation similarity, we take *saleslady* (Figure 1) and *conductor* (Figure 2) for example (Li, 2012) which are similar on morphology. When computing a pair of nodes similarity, such as root nodes, they are regarded as current calculating nodes (CN). Then both of CN themselves and the children nodes of CN are taken into consideration. CN (*human*) of *saleslady* has relations of *hostof*, *domain*, *modify* and *none* (no relation). With the same relations in CN (*human*) of *conductor*, we get the similarity of children nodes as one relation similarity of a pair of CN. In other words, the similarity of children nodes which have the same relation with their respective father nodes will be computed. If there is no match, the relation similarity is defaulted as small constant  $\delta$ . Every pair of nodes should be calculated in DEF tree in the same layer as formula (1).

$$Sim_{node}(S_1, S_2) = \beta_{rela} \frac{1}{N} \sum_{i=1}^N Sim_{rela\_i}(S_1, S_2) + \beta_s Sim_s(S_1, S_2) \quad (1)$$

Where,  $N$  denotes  $N$  different kinds of relations,  $Sim_{rela\_i}(S_1, S_2)$  denotes the  $i$ -th relation similarity which in fact expresses the children node similarity of the pair  $(S_1, S_2)$ ,  $Sim_s(S_1, S_2)$  denotes sememe similarity, and  $\beta_{rela} \geq 0$ ,  $\beta_s \geq 0$ ,  $\beta_{rela} + \beta_s = 1$ . Bottom-to-up algorithm will be used to recursively compute DEF similarity in order to achieve the root node similarity as the DEF similarity.

$$Sim_{DEF}(S_1, S_2) = Sim_{node}(S_1, S_2) \text{ if } S_1 = root1, S_2 = root2$$

The key point of DEF similarity computing method is not only taking the migration process of the nodes in the DEF tree into consideration (Li, 2012), but also using the relation between children nodes and their respective father node. In this way, the structure information from the DEF tree can be fully exploited.

However, there are special sememe (Attribute Sememe and Secondary Feature Sememe) whose weights are so high that the similarity unreasonably increases. Therefore, the formula (2) derived from the formula (1) is used to compute node similarity with a penalty factor  $\varepsilon$ .

$$Sim_{node}(S_1, S_2) = \beta_{rela} \frac{1}{N} \sum_{i=1}^N Sim_{rela\_i}(S_1, S_2) + \varepsilon \cdot \beta_s Sim_s(S_1, S_2) \quad (2)$$

### 2.2 Sememe similarity computing

The sememes are also described by DEF in HowNet2008. Therefore, sememe similarity ( $Sim_s(S_1, S_2)$ ) can be divided into two parts, namely structure similarity and DEF similarity.

#### 2.2.1 Structure similarity between sememes

In related works, many features of tree, such as the distance, depth and the least common nodes (LCN) in tree, have been used. This paper uses formula (3) below to compute structure similarity of sememe similarity

$$StructSim(S_1, S_2) = \frac{\alpha \cdot (\text{depth}(S_1) + \text{depth}(S_2))}{\alpha \cdot (\text{depth}(S_1) + \text{depth}(S_2)) + \text{dist}(S_1, S_2) + |\text{depth}(S_1) - \text{depth}(S_2)|} \quad (3)$$

Where,  $\text{depth}(S_1)$  represents depth of  $S_1$  in sememe tree, and  $\text{dist}(S_1, S_2)$  is the distance between  $S_1$  and  $S_2$  in sememe tree. It is clear that structure similarity of sememes increases with shorter distance between the sememes and a smaller difference in depth.

In the process of computing structure similarity, if there exists an antonym relation or converse relation between  $S_1$  and  $S_2$ , or so does the same relation in the path between  $S_1$  and  $S_2$  in sememe tree, mark a flag with “-”. However, antonym

relation and converse relation listed in HowNet document are too strict. Synonym Dictionary is used to extend antonym and converse relation.

### 2.2.2 DEF similarity between sememes

A special phenomenon exists in two aspects. On the one hand, in the process of computing DEF similarity, sememe similarity computing is needed. On the other hand, in the process of computing sememe similarity, DEF similarity computing is needed. This phenomenon brings about a cyclical calculation. In order to terminate infinite cyclical calculation, cyclical calculation will be processed only twice using formula (4)

$$Sim_s(S_1, S_2) = \begin{cases} StructSim(S_1, S_2) & \text{if last circle} \\ \beta_{struct} StructSim(S_1, S_2) + \beta_{DEF} Sim_{DEF}(S_1, S_2) & \text{if not last circle} \end{cases} \quad (4)$$

Where,  $StructSim(S_1, S_2)$  denotes structure similarity, and  $\beta_{struct} \geq 0$ ,  $\beta_{DEF} \geq 0$ ,  $\beta_{struct} + \beta_{DEF} = 1$ .  $Sim_{DEF}(S_1, S_2)$  equals 1 if there is no DEF description of sememe in both  $S_1$  and  $S_2$ . Convergence with cyclical calculation instead of twice will be researched in our future work.

### 2.3 Word similarity computing

Formula (5) below will be used to compute similarity between words containing one or more DEF description by

$$Sim_w(W_1, W_2) = \pm \max_{i=1..n, j=1..m} |Sim_{DEF}(S_{1i}, S_{2j})| \quad (5)$$

Where,  $S_{1i}$  is the  $i$ -th DEF of word  $W_1$ ,  $S_{2j}$  is the  $j$ -th DEF of word  $W_2$ , “+” and “-” depend on the flag (section 2.2.1) of max DEF similarity. In formula (5), we choose maximum DEF similarity as word similarity by default.

## 3 Experiment and Comparison

General parameters in experiments derive from Liu’s and Li’s. The special parameters are optimized with greedy algorithm. Table 1 gives all the parameters of experiment.

### 3.1 Nouns and Verbs experiment

The result of our approach contrasted with Liu’s and Li’s is shown in Table 2. In Liu’s approach,

general parameter	$\alpha$	$\delta$	$\beta_{rela}$	$\beta_s$
value	1.6	0.1	0.3	0.7
special parameter	$\beta_{struct}$	$\beta_{DEF}$	$\varepsilon$	
value	0.4	0.6	0.1	

Table 1. Parameters of experiment

the similarity between words, such as pair of “man” and “father” and pair of “pink” and “crimson”, is unreasonable. Our algorithm performs as well as Li’s in solving this problem. What’s more, through adding flag to mark antonym relation, our algorithm performs better than Li on some pairs of words, such as “man” and “woman” with a flag “-” marking antonym.

### 3.2 Adjectives experiment

Li’s algorithm and Liu’s algorithm never take antonym relation into consideration. Jiang (2008) extends Liu’s algorithm by using antonym relation. Table 3 shows that our result is much better than Jiang’s result in many words. As we know, “beautiful” and “shifty-eyed” is strictly a pair of antonyms, and “shifty-eyed” is “ugly” but not vice versa.

Word 1	Word 2	Liu’s result	Li’s result	Our result
男人 (man)	女人 (woman)	0.8611	0.8955	-0.9957
男人 (man)	父亲 (father)	1.0000	0.8902	0.8904
男人 (man)	母亲 (mother)	0.8611	0.7857	-0.8875
粉红色 (pink)	深红色 (crimson)	1.0000	0.8500	-0.9829
名声 (reputation)	硬度 (hardness)	0.6176	/	0.2585
三伏 (hot)	冬眠 (hibernate)	0.0429	/	-0.6555

Table 2. Comparison of nouns and verbs

Word 1	Word 2	Jiang’s result	Our result
美丽 (beautiful)	丑陋 (ugly)	-1.0000	-1.0000
美丽 (beautiful)	贼眉鼠眼 (shifty-eyed)	-1.0000	-0.9662
美丽 (beautiful)	优雅 (elegant)	0.7884	0.9264
舒服 (comfortably)	残疾 (handicap)	-0.0664	-0.7989
勇敢 (brave)	坚强 (strong)	0.7884	0.9500

Table 3. Comparison of adjectives

### 3.3 Synonyms experiment

In synonyms experiment, nearly 8000 pairs of words are randomly chosen as experimental data. The result (Figure 3) illustrates the effectiveness of our approach, since most of synonyms similarity is very high. Table 4 shows that our approach performs better than Li’s in computing similarity of synonyms.

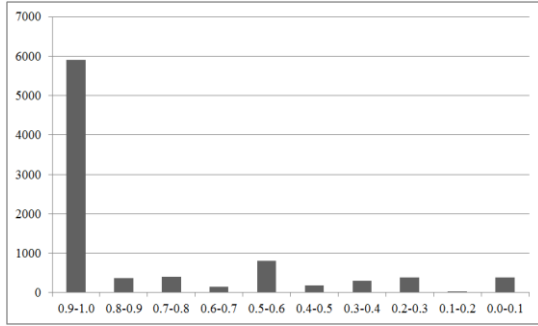


Figure 3. Result of synonyms experiment

	sim>0.9	sim>0.8	sim>0.7
Li's approach	60.89%	68.75%	72.85%
Our approach	66.05%	70.21%	74.76%

Table 4. Percentage of synonyms in different ranges

### 3.4 Antonyms experiment

Nearly 3000 pairs of antonyms are crawled from web resource for experiment. And the experimental results of antonyms come from two parts. One is the absolute value of antonyms experimental result denoting antonymous degree that is shown in Figure 4, and the other one is the flag “-” (section 2.2.1) marking antonym. Table 5 shows the percent of antonym in different ranges of similarity. Table 6 shows the number of pairs of antonyms with flag “-” by our approach.

The experimental results prove the high effectiveness of our approach of computing word similarity for most of antonyms similarity. However, it performs not very well in finding the flag “-” which marks antonym. With the development of HowNet, our approach will perform better.

Absolute similarity	>0.9	>0.8	>0.7
	50.02%	61.89%	68.70%

Table 5. Percentage of antonyms in different ranges

method	number in 3000 pairs
Original	863
Extend-Antoymys	966

Table 6. Number of antonyms with flag “-”

### 3.5 SemEval experiment

The datasets of Evaluating Chinese Word Similarity task In SemEval 2012 is used as the experimental data, of which the values are normalized as [0, 1]. The experimental data (130 pair words) covers similarity ranging from 0 to 1. Experimental data are sequenced by their similarity

from high to low. The result of experiments is shown in Figure 5.

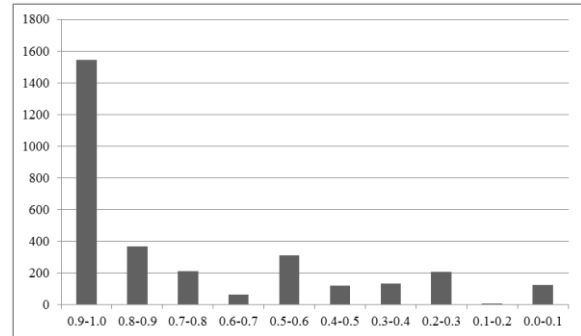


Figure 4. Result of antonyms experiment

Compared with Liu's method, the result shows that in the pairs of high similarity words, the difference of similarity is nearly 0.095. Besides, the largest difference is lower than 0.1. In Figure 5, the low difference value (0.01) between the highest difference and lowest difference is verified that the approach proposed by this paper is effective and stable in different range of similarity.

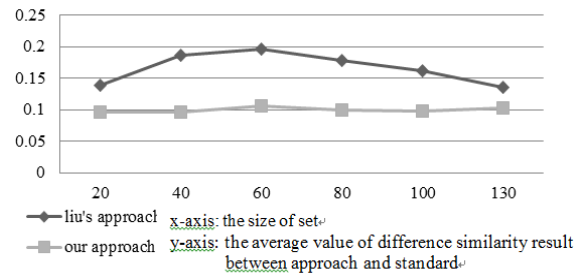


Figure 5. Comparison of experimental results

## 4 Conclusions and Future work

This paper proposes a new approach for computing word similarity between Chinese words using HowNet. The contribution can be concluded below. (1) Improve the accuracy of similarity by using EF description in sememe hierarchy; (2) substantiate that different kinds of sememe describe DEF in different weight; (3) use the Synonym Dictionary to alleviate strict limitations in antonym and converse relation.

Due to the importance of word context, in future, for documents, a method to choose suitable DEF for the word is necessary depending on context instead of maximum DEF similarity. Moreover, the alignment between sub-description of DEF is meaningful in computing semantic similarity. We will pay extra attention to sub-tree alignment. Based on these, we will optimize parameters for various applications.

## References

- Razvan Bunescu and Yunfeng Huang. (2010b). A utility driven approach to question ranking in social QA. In *Proceedings of The 23rd International Conference on Computational Linguistics (COLING 2010)*, 125–133.
- Michael A.G. Mohler, Razvan Bunescu and Rada Mihalcea. (2011). Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency GraphAlignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 752–762.
- Lan Wang and Yuan Wan. (2011). Sentiment Classification of Documents Based on Latent Semantic Analysis. In *Communications in Computer and Information Science*, (176), 356-361
- Qun Liu and Sujian Li, (2002) Word Semantic Similarity Computing Based on HowNet, *Computational Linguistics and Chinese Language Processing*, (7): 59-76.
- Bin Ge, Fangfang Li, Silu Guo and Daquan Tang. (2010). The Research on Lexical Semantic Similarity Computing based on HowNet[J]. *Application Research of computers*, 27(9): 3329-3333
- Preslav Nakov and Marti A. Hearst (2008). Solving relational similarity problems using theweb as a corpus. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 452-460
- Kazuhiro Morita, Shuto Arai, Hiroya Kitagawa, Masao Fuketa and Jun-ichi Aoe. (2011) Dynamic Construction of Hierarchical Thesaurus using Cooccurrence Information. *The 2nd International Conference on Networking and Information Technology IPCSIT*, Singapore
- Roberto Navigli and Simone Paolo Ponzetto. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.
- Dmitry Davidov and Ari Rappoport (2010). Automated Translation of Semantic Relationships. In *Proceedings of The 23rd International Conference on Computational Linguistics (COLING 2010)*, 241-249.
- Zhengdong Dong and Qiang Dong. (2002) Introduction to HowNet, <http://www.keenage.com>.
- Hua Li, Changle Zhou, Min Jiang, Ke Cai. (2012). A hybrid approach for Chinese word similarity computing based on HowNet. *Automatic Control and Artificial Intelligence (ACAI 2012)*, 80-83
- Peng Jin, Yunfang Wu. (2012). SemEval-2012 task 4: evaluating Chinese word similarity. In *Proceeding of the First Joint Conference on Lexical and Computational Semantics*. (1): 374-377
- Min Jiang, Shibin Xiao, Hongwei Wang and Shuicai Shi. (2008). A improved Semantic Similarity Computing based on HowNet[J]. In *Journal of Chinese information processing*, 22(5): 84-89
- Zhengdong Dong and Qiang Dong. (2006) HowNet and the Computation of Meaning, *World Scientific Publishing*.
- Feng Li, Fang Li. (2007) An New Approach Measuring Semantic Similarity in Hownet 2000, *Journal of Chinese Information processing*.



# Extracting Causes of Emotions from Text

**Alena Neviarouskaya**

Toyohashi University of Technology  
Toyohashi, Japan  
alena@kde.cs.tut.ac.jp

**Masaki Aono**

Toyohashi University of Technology  
Toyohashi, Japan  
aono@kde.cs.tut.ac.jp

## Abstract

This paper focuses on the novel task of automatic extraction of phrases related to causes of emotions. The analysis of emotional causes in sentences, where emotions are explicitly indicated through emotion keywords can provide the foundation for research on challenging task of recognition of implicit affect from text. We developed a corpus of emotion causes specific for 22 emotions. Based on the analysis of this corpus we introduce a method for the detection of the linguistic relations between an emotion and its cause and the extraction of the phrases describing the emotion causes. The method employs syntactic and dependency parser and rules for the analysis of eight types of the emotion-cause linguistic relations. The results of evaluation showed that our method performed with high level of accuracy (82%).

## 1 Introduction and Background

Emotional reactions to three salient aspects of the world, namely (1) events and their consequences, (2) agents and their actions, and (3) objects, are based on the nature of cognitive origins and can be triggered under specific conditions (Ortony et al., 1988). The cognitive model of emotions (OCC model of emotions) arranges 22 emotions in three substantially independent classes according to the aspects of the world that are in focus of evaluation.

Recently, the task of automatic recognition of distinct emotions conveyed in text has been gaining increased attention of researchers in the areas of natural language processing and computational linguistics (Alm, 2008; Aman and Szpakowicz, 2008; Boucouvalas, 2003; Chaumartin, 2007; Katz et al., 2007; Kozareva et al., 2007; Liu et al., 2003; Neviarouskaya et al., 2011; Purver and

Battersby, 2012; Strapparava and Mihalcea, 2008; Suttles and Ide, 2013). To understand emotions expressed in written language, it is important to analyse the causes of emotions ("*what caused a particular emotion*") and eliciting conditions ("*under what conditions*"). The challenge of emotion cause detection in text has been recently tackled by Chen et al. (2010), who developed two sets of linguistic pattern-based features (manually generalized patterns and automatically generalized patterns) for extraction of causes for emotions in Chinese. The linguistic-pattern-based methodology described in (Chen et al., 2010) inspired the development of a method for the identification of Italian sentences that contain emotion cause phrases and the retrieval of emotion – emotion cause phrase couples (Russo et al., 2011). In their subsequent work, Caselli et al. (2012) semi-automatically assigned polarity values to Italian nouns that potentially represent nominal cause events associated with emotions.

In this work, we introduce a novel method for automatic extraction of emotion causes. The main contributions of our work are as follows: (1) development of a corpus of emotion causes and (2) deep analysis of cause events specific for 22 emotions from the OCC model. The analyses of emotional causes in sentences, where emotions are explicitly indicated through emotion keywords, and conditions that lead to emotional experience can provide the foundation for research on challenging task of recognition of implicit affect from text.

## 2 Development and Analysis of the Corpus of Emotion Causes

### 2.1 Creation of the Dataset of Sentences with Explicitly Indicated Emotions

In the text of (Ortony et al., 1988), about 130 tokens (emotion words) have been distributed between 22 emotion types. For example, '*glad*' and

'happy' correspond to *Joy* emotion class; 'scared' and 'terrified' are associated with *Fear* emotion; and 'awe' and 'esteem' describe *Admiration* emotion. We consider these tokens as seed terms for extraction of sentences that contain information on what caused the particular emotion.

In addition to 22 sentences provided in (Ortony et al., 1988) as examples for each emotion type, we manually collected 510 sentences with emotion tokens and explicitly mentioned emotion causes from online ABBYY Lingvo dictionary (<http://www.lingvo-online.ru/en>). 118 emotion tokens were found productive, resulting in at least one cause-containing sentence per emotion token.

The corpus consisting of 532 sentences was manually annotated. The annotation task included the following subtasks: (1) to define an agent or an experiencer of emotion specified by emotion token; (2) to delimit the phrase describing the cause of emotion; (3) to define the linguistic relation between emotion and its cause; (4) to classify the cause event as positive, negative, or neutral; and (5) to extract tokens that influence the polarity of the phrase.

## 2.2 Corpus Analysis

We performed the detailed analysis of the created corpus. The agent or experiencer of emotion specified by emotion token was defined in 495 sentences (93% from the whole corpus). In the corpus, about 46% of sentences are related to positive emotions, and about 54% of sentences express negative emotions.

The analysis of polarity of cause events from the annotated corpus showed the following distribution of the causes according to the sentiment categories: (1) positive – about 27%; (2) negative – about 29%; and (3) neutral – about 44% of the cause events. These figures emphasize the fact that the cause of emotion expressed in text is not necessarily described by sentiment words. Interesting observation is that cause events are negative in 2.9% of sentences with positive emotions, and positive cause events occur in 4.5% of sentences with negative emotions (for example, '*And people changed from diet to diet and felt guilty [negative emotion] because they continued to like the things they weren't supposed to*').

The important feature that was identified in each sentence was the linguistic relation between emotion and its cause. Based on the analysis of the annotated data, we distinguish eight types of such linguistic relations:

1. One-word preposition (OWP). For example, 'at' in the sentence '*And while she gaped*

*with disappointment at his lukewarmness, he got himself away, at ten*'.

2. Complex preposition (CP). For example, 'because of' in the sentence '*He was himself a Greek, and there were many who felt offended because of his height*'.

3. Coordinating conjunction (CC). For example, 'for' in the sentence '*La Cote was much depressed, for he had scored here the worst failure of his campaign*'.

4. Subordinating conjunction (SC). For example, 'because' in the sentence '*And people changed from diet to diet and felt guilty because they continued to like the things they weren't supposed to*'.

5. Subject (SUBJ). For example, in the sentence '*His tone scared her more than anything she could remember*', the subject 'his tone' represents the cause of *Fear* emotion expressed by the verb 'scared'.

6. Verb or predicate (V). For example, the predicate 'filled with' connects the *Joy* emotion with its cause in the sentence '*As for the captain, the presence in his room of the children, who came to cheer up Ilusha, filled his heart from the first with ecstatic joy*'.

7. Object (OBJ). For example, in the sentence '*I adore poetry*', the object 'poetry' triggers *Love* emotion that is reflected through the verb 'adore'.

8. Attributive nominal (ATT). For example, in the sentence '*It is a sad tale, a very sad tale*', emotional adjective 'sad' describes the noun 'tale' through attributive nominal relation (in this sentence, 'tale' causes *Distress* emotion).

In Table 1, the specific emotion-cause linguistic relations that were found in our corpus of sentences are listed according to their frequency. One-word prepositions (including 'to', 'for', 'of', 'at', 'with', 'by', 'about', 'over' etc.) acting as linkages between emotion tokens and phrases describing the cause of emotion occur in about 68.2% of sentences. Subordinating conjunctions (examples include 'that', 'when', 'because', 'as' etc.) constitute about 21.4% of sentences. The object and subject are the next frequent relation types (about 6% and 2.3% of sentences, respectively).

## 3 Method for Extraction of Emotion Causes

Our method for automatic extraction of emotion causes is based on the analysis of syntactic and dependency information from the parser. In our

Relation	Type	Frequency (number)	Frequency (%)
<i>to</i>	OWP	77	14.47
<i>for</i>	OWP / CC	73	13.72
<i>that</i>	SC	63	11.84
<i>of</i>	OWP	48	9.02
<i>at</i>	OWP	42	7.89
<i>with</i>	OWP	37	6.95
object	OBJ	32	6.02
<i>by</i>	OWP	25	4.70
<i>about</i>	OWP	22	4.14
<i>when</i>	SC	21	3.95
<i>over</i>	OWP	20	3.76
<i>because</i>	SC	15	2.82
subject	SUBJ	12	2.26
<i>in</i>	OWP	9	1.69
<i>on</i>	OWP	7	1.32
attribute	ATT	6	1.13
<i>as</i>	SC	5	0.94
<i>if</i>	SC	5	0.94
<i>as though</i>	SC	4	0.75
<i>filled with; fostered by; trigger</i>	V	3	0.56
<i>after</i>	OWP / SC	1	0.19
<i>as if</i>	SC	1	0.19
<i>because of</i>	CP	1	0.19
<i>from</i>	OWP	1	0.19
<i>under</i>	OWP	1	0.19
<i>without</i>	OWP	1	0.19

Table 1. Emotion-cause linguistic relations and their frequency in the corpus

work we employ Connexor Machine Syntax (<http://www.connexor.eu/technology/machinese/>) that is applied to each sentence in order to get lemmas, dependencies, syntactic and morphological information (see example in Table 2). Using parser output, the method extracts phrases that characterize the emotion causes.

The algorithm detects and extracts cause phrases introduced by prepositions (OWP and CP) through three rules:

1. POSTMODIFIER rule: if morphological tag of the cause marker is *PREP* and this preposition is linked with the emotion token through *mod* syntactic relation, then extract all tokens related to this preposition.

2. NEXT TOKEN rule: if morphological tag of the cause marker is *PREP* and syntactic relation of this preposition is unavailable (*null* relation), then if this cause marker directly follows the emotion token, extract all tokens related to this preposition.

3. VERB-MEDIATED RELATION rule: if morphological tag of the cause marker is *PREP* and this preposition is directly connected with

Id	Token	Lemma	Dependency	Tags
1	Most	many	qn:>2	@QN> %>N DET SUP PL
2	doctors	doctor	subj:>3	@SUBJ %NH N NOM PL
3	are	be	v-ch:>4	@+FAUXV %AUX V PRES
4	attracted	attract	main:>0	@-FMAINV %VP EN
5	to	to	ha:>4	@ADVL %EH PREP
6	medicine	medicine	pcomp:>5	@<P %NH N NOM SG
7	because	because	pm:>9	@CS %CS CS
8	they	they	subj:>9	@SUBJ %NH PRON PERS NOM PL3
9	look	look	cnt:>4	@+FMAINV %VA V PRES
10	forward	forward	goa:>9	@ADVL %EH ADV
11	to	to	ha:>9	@ADVL %EH PREP
12	curing	cure	pcomp:>11	@<P-FMAINV %VA ING
13	disease	disease	obj:>12	@OBJ %NH N NOM SG

Table 2. Example of parser output

verb, to which emotion token is related within the clause, and the id of preposition is higher than that of emotion token, then extract all tokens related to this preposition.

The rules for extraction of phrases connected to emotion tokens through conjunctions (SC and CC) are as follows:

1. THAT rule: if morphological tag of the '*that*' cause marker is *CS* and the id of conjunction is higher than that of emotion token, then if verb of subordinate clause, to which the conjunction '*that*' is connected, is related to emotion token through chain of relations, extract all tokens related to the verb of subordinate clause.

2. DEPENDENT CLAUSE rule: if morphological tag of the cause marker is *CS* or *CC*, and the dependent verb, to which conjunction is related, is connected to the main verb, to which emotion token is related (here, the emotion token might be the verb itself), then extract all tokens related to the verb of dependent clause.

To detect verbs for the above rules, the algorithm looks for the following functional tags: @+FMAINV (finite main verb), @-FMAINV (nonfinite main verb), and @<P-FMAINV (nonfinite clause as preposition complement).

The extraction of emotion causes represented by either subject (SUBJ), or predicate (V), or object (OBJ), or attributive nominal (ATT) linguistic relations is based on the analysis of *subj*, *obj*, and *att* syntactic relations and the corresponding tokens.

## 4 Evaluation

Based on the emotion cause phrases extracted by human annotator from our corpus consisting of 532 sentences, we evaluated the appropriateness of the phrases extracted by our algorithm. In each pair of phrases, the number of words was calculated (namely, number of gold standard tokens and number of automatically extracted tokens). Then, the number of words correctly extracted by our algorithm was found, and we calculated precision, recall, and F-score for each automatically extracted phrase. The results averaged over all the phrases are given in Table 3 (including the results on different groups and all emotion cause linguistic relations).

Emotion cause linguistic relations	Accuracy of phrase extraction		
	Precision	Recall	F-score
Prepositions (OWP, CP)	0.715	0.723	0.700
Conjunctions (SC, CC)	0.470	0.549	0.473
Subject, predicate, object, and attributive nominal (SUBJ, V, OBJ, ATT)	0.787	0.793	0.772
All relations	<b>0.670</b>	<b>0.692</b>	<b>0.658</b>
All relations (after improving the method based on error analysis)	<b>0.821</b>	<b>0.852</b>	<b>0.810</b>

Table 3. Evaluation of the appropriateness of automatically extracted emotion causes

As seen from the obtained results, our algorithm achieved the highest level of precision (0.787) in extracting emotion cause phrases represented by subject, predicate, object, and attributive nominal linguistic relations, while it was least precise (0.470) in case of emotion causes introduced by conjunctions. We obtained good results considering all emotion cause linguistic relations: precision in 0.670, recall in 0.692, and F-score in 0.658.

We performed an error analysis on the sentences, where our method failed to extract correct phrases. The classification and distribution of errors is given in Table 4. In most cases (about 44.8%), the method failures were due to missing rule for infinitive marker 'to' (morphological tag *INFMARK*>, in contrast to preposition tag *PREP*). For example, 'to' in the sentence *In that regard, New Zealand is proud to work towards nuclear disarmament with the other members of the New Agenda Coalition*. About 22.4% of errors were caused by inability of the parser to output correct tags for syntactic relations. Analysis of 'when' as a relative adverb (*ADV* and *WH* morphological tags), in addition to it as a subor-

Error type	Frequency (number)	Frequency (%)
Infinitive marker 'to'	60	44.78
Null or incorrect tag from parser	30	22.39
'When' as a relative adverb	18	13.43
Missing subordinating conjunction 'that'	11	8.21
THAT rule	4	2.99
POSTMODIFIER rule	3	2.24
Emotion phrase 'look forward'	3	2.24
Reference resolution	3	2.24
Coordinating conjunction in SUBJ and OBJ rules	2	1.5
<b>Total</b>	<b>134</b>	<b>100</b>

Table 4. Classification and distribution of errors

ordinating conjunction, would deal with about 13.4% of errors. We found that the emotion causes represented by subordinate clauses without such a marker of subordination as 'that' pose the main challenge, as the parser outputs *null* relations for such dependent clauses (for example, clause *I never had to lie then* in the sentence *I reckon I was so glad I never had to lie then*). The analysis of errors showed the necessity to improve several rules (such as THAT, POSTMODIFIER, SUBJ, and OBJ rules). The method would also benefit from adding reference resolution. For example, using reference resolution, the method could extract *these difficulties* instead of *they* as emotion cause from the sentence *I could not dwell upon these difficulties fully, for they made me far too uneasy*.

After improving the emotion cause extraction method by adding and modifying the rules, we obtained the following evaluation results: precision in 0.821, recall in 0.852, and F-score in 0.810 (last row in Table 3). In that way, our method performed with about 15% gain in accuracy.

## 5 Conclusions

The main contributions of our work are the creation of a corpus of emotion causes specific for 22 emotions from the OCC model and the development of a novel method for extraction of emotion causes from sentences based on the analysis of syntactic and dependency information provided by the parser. In future research we plan to improve our emotion cause extraction method and incorporate the automatic detection of an experiencer of emotion specified by emotion token and the classification of causes as positive, negative, or neutral.

## References

- Andrew Ortony, Gerald L. Clore, and Allan Collins. 1988. *The Cognitive Structure of Emotions*. Cambridge University Press.
- Cecilia O. Alm. 2008. *Affect in Text and Speech*. PhD Dissertation, Urbana, IL: University of Illinois at Urbana-Champaign.
- Saima Aman and Stan Szpakowicz. 2008. Using Roget's Thesaurus for Fine-Grained Emotion Recognition. *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pp. 296-302.
- Anthony C. Boucouvalas. 2003. Real Time Text-to-Emotion Engine for Expressive Internet Communications. In *Being There: Concepts, Effects and Measurement of User Presence in Synthetic Environments*, Ios Press, pp. 306-318.
- Francois-Regis Chaumartin. 2007. UPAR7: A Knowledge-Based System for Headline Sentiment Tagging. *Proceedings of SemEval-2007*.
- Phil Katz, Matt Singleton, and Richard Wicentowski. 2007. SWAT-MP: the SemEval-2007 Systems for Task 5 and Task 14. *Proceedings of SemEval-2007*.
- Zornitsa Kozareva, Borja Navarro, Sonia Vazquez, and Andres Montoyo. 2007. UA-ZBSA: A Headline Emotion Classification through Web Information. *Proceedings of SemEval-2007*.
- Hugo Liu, Henry Lieberman, and Ted Selker. 2003. A Model of Textual Affect Sensing Using Real-World Knowledge. *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 125-132.
- Matthew Purver and Stuart Battersby. 2012. Experimenting with Distant Supervision for Emotion Classification. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 482-491.
- Carlo Strapparava and Rada Mihalcea. 2008. Learning to Identify Emotions in Text. *Proceedings of the 2008 ACM Symposium on Applied Computing*, pp. 1556-1560.
- Jared Suttles and Nancy Ide. 2013. Distant Supervision for Emotion Classification with Discrete Binary Values. *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 121-136.
- Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2011. Affect Analysis Model: Novel Rule-Based Approach to Affect Sensing from Text. *International Journal of Natural Language Engineering*, 17(1):95-135. Cambridge University Press.
- Ying Chen, Sophia Y. M. Lee, Shoushan Li, and Churen Huang. 2010. Emotion Cause Detection with Linguistic Constructions. *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 179-187.
- Irene Russo, Tommaso Caselli, and Francesco Rubino. 2011. EMOCause: An Easy-Adaptable Approach to Emotion Cause Contexts. *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT 2011*, pp. 153-160.
- Tommaso Caselli, Irene Russo, and Francesco Rubino. 2012. Assigning Connotation Values to Events. *Proceedings of the Eight International Conference on Language Resources and Evaluation*, pp. 3082-3089.

# Automated Grammar Correction Using Hierarchical Phrase-Based Statistical Machine Translation

**Bibek Behera, Pushpak Bhattacharyya**  
Dept. of Computer Science and Engineering  
IIT Bombay, Mumbai, India  
{bibek, pb}@cse.iitb.ac.in

## Abstract

We introduce a novel technique that uses hierarchical phrase-based statistical machine translation (SMT) for grammar correction. SMT systems provide a uniform platform for any sequence transformation task. Thus grammar correction can be considered a translation problem from incorrect text to correct text. Over the years, grammar correction data in the electronic form (i.e., parallel corpora of incorrect and correct sentences) has increased manifolds in quality and quantity, making SMT systems feasible for grammar correction. Firstly, sophisticated translation models like hierarchical phrase-based SMT can handle errors as complicated as reordering or insertion, which were difficult to deal with previously through the mediation of rule based systems. Secondly, this SMT based correction technique is similar in spirit to human correction, because the system extracts grammar rules from the corpus and later uses these rules to translate incorrect sentences to correct sentences. We describe how to use Joshua, a hierarchical phrase-based SMT system for grammar correction. An accuracy of 0.77 (BLEU score) establishes the efficacy of our approach.

## 1 Introduction

We consider grammar correction as a translation problem - translation from an incorrect sentence to a correct sentence. The correcting system is trained using a parallel corpus of incorrect and their corresponding correct sentences. The system learns SCFG (synchronous context free grammar) rules (Chiang, 2005) during translation. SCFG rules look like this:-

- $X \rightarrow X_1 \text{ of } X_2, X_1 \text{ for } X_2$

The above rule implies that phrases  $X_1$  and  $X_2$  in source language are translated to phrases in target language, while *of* is replaced with *for*. The position of both phrases w.r.t. *of* remains same in the target language, which means there is no reordering of phrases.

After generating such grammar rules, it converts the erroneous sentence to a tree using the rules of grammar, i.e., the left hand side of the SCFG rules. It then applies correction rules, i.e., the right hand side of the SCFG rules, to convert the tree as explained later in section 3. The yield of the tree generates the corrected sentence.

Here are various types of errors that one encounters in grammar correction:

Article choice errors:- *a Himalayas is the longest mountain range in the world.* The correct translation is '*the Himalayas is the longest mountain range in the world*'.

Preposition errors:- *Helicopter crashed at central London.* The correct translation is '*Helicopter crashed in central London*'.

Word form errors:- *The rain mays fall in July* should be changed to '*The rain may fall in July*'.

Word insertion errors:- *The court deemed necessary that she respond to the summons* should be changed to '*The court deemed it necessary that she respond to the summons*'.

Reordering errors:- *never we miss deadlines* should be corrected to '*we never miss deadlines*'.

Article choice errors and preposition errors have been tackled by rule based techniques. But rules are customized, so to say, for each error, which is a time consuming and fragile process. SMT, on the other hand, treats all errors uniformly, considering error correction as a translation problem. Secondly, problems such as reordering or word insertion are well known in machine translation.

The roadmap of the paper is as follows. In section 2, we discuss previous work. In section 3, we elaborate on how hierarchical machine translation system can do automatic grammar correction. Section 4 states the grammar rules that are extracted by the system automatically. In section 5, we present our experiments followed by the results in section 6. We conclude in section 7 with pointers to future work.

## 2 Background

Initially the work that has been done in grammar correction is based on identifying grammar errors. Chodorow and Leacock (2000) used an ngram model for error detection by comparing correct ngrams with ngrams to be tested. Later, classification techniques like Maximum entropy models have been proposed (Izumi et al., 2003; Tetreault and Chodorow, 2008; Tetreault and Chodorow, 2008). These classifiers not only identify errors, but also correct them. These methods do not make use of erroneous words thus making error correction similar to the task of filling empty blanks. While in editing sentences, humans often require the information in the erroneous words for grammar correction.

In other works, machine translation has been previously used for grammar correction. Brockett (2006) used phrasal based MT for noun correction of ESL students. Désilets and Hermet (2009) translate from native language L1 to L2 and back to L1 to correct grammar in their native languages. Mizumoto (2012) also used phrase-based SMT for error correction. He used large-scale learner corpus to train his system. These translation techniques suffered from lack of good quality parallel corpora and also good translation systems.

If high quality parallel corpus can be obtained, the task of grammar correction becomes easy using a powerful translation model like hierarchical phrase based machine translation.

## 3 Automatic grammar correction using hierarchical phrase-based SMT

In this section we discuss the working and the implementation of the grammar correction system.

### 3.1 Working

Grammar correction can be seen as a process of translation of incorrect sentences to correct ones. Basically the translation system needs a parallel

corpus of incorrect and correct sentences. The system starts with an alignment to obtain word to word translation probabilities. The second stage is grammar extraction using the hiero style of grammar (Chiang, 2005). Non-terminals are generalized form of phrases, *i.e.*, all possible phrases allowed in the framework of Chiang (2005) are represented by the symbol  $X$ . There is another symbol  $S$  to start the parse tree. These rules are in the form of SCFG rules. If the incorrect sentence is, ‘*few has arrived*’ and the correct sentence is, ‘*few have arrived*’, the grammar rules extracted are :-

- $X \rightarrow \text{few has } X_1, \text{ few have } X_1$
- $X \rightarrow \text{arrived}, \text{ arrived}$

The first rule means that *few has* followed by a phrase may be translated to *few have* followed by translation of that phrase. Second rule suggests that any phrase that yields *arrived* can be translated to *arrived*.

After the grammar extraction is done, the left sides of the grammar rules are stripped and used to generate the parse tree of the sentence *few has arrived*.

Here are the left side rules:-

- $X \rightarrow \text{few has } X_1$
- $X \rightarrow \text{arrived}$

Also, there is a “glue rule” to combine two trees or just derive a non terminal from the start symbol  $S$ .

- $S \rightarrow S^1 X \mid X$

The glue rule is used to start the parsing process. It generates a sub-tree for the string *few has* and a non-terminal for *arrived*. Then the right side rules are used to convert *few has* to *few have* as shown in Figure 1, while *arrived* is translated as *arrived*.

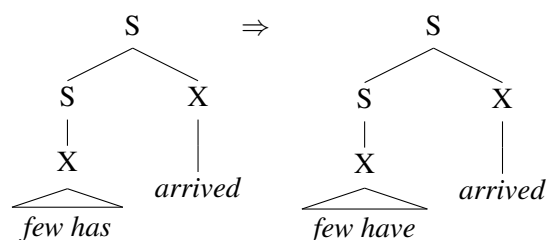


Figure 1: Parse tree for transformation from incorrect to correct sentences.

<sup>1</sup>Here S means start of the tree

The yield of the tree generates *few have arrived*, which is the required correction. This is the essence of decoding in hierarchical machine translation.

### 3.2 Implementation

The translation system being used is the Joshua Machine translation system (Li et al., 2010). The SMT based correction pipeline is a six step process in conformity with the Joshua decoder (Ganitkevitch et al., 2012). First we create the dataset in a input folder with six files such as:-

1. train.incorrect- Incorrect sentences in our training corpus
2. train.correct- Correct sentences in our training corpus
3. tune.correct- Incorrect sentences in our development set
4. tune.incorrect- Correct sentences in our development set
5. test.correct- Incorrect sentences in our testing set
6. test.incorrect- Correct sentences in our testing set

The pipeline starts with preprocessing the corpus, *i.e.*, tokenisation and lowercasing followed by word alignment. The result of word alignment is stored in training.align file. Then a file, “grammar.gz” is created by joshua that stores SCFG rules using information from the training.align file and the training corpus. This process is called grammar generation and is followed by the building of the language model.

For developing the language model, the Joshua MT system uses KenLM (Heafield, 2011) toolkit or BerkeleyLM. This is the end of the training process. The steps that follow in the pipeline are tuning and testing. Tuning iterates over the development set to obtain the best parameters for the translation model. At the end of tuning, the system obtains the optimized parameters that can be deployed into the translation model for testing. The testing phase translates sentences from test set to evaluate the overall BLEU score (Papineni et al., 2002).

## 4 Analysis of grammar rules extracted

In this section we look at how various grammar corrections have been handled. The various types of errors handled are article choice errors, preposition errors, word-form choice errors and word insertion errors as mentioned in Park and Levy (2011). Apart from these errors, we also discuss errors due to reordering and errors due to unseen verbs which have not been implemented in previous models.

### 4.1 Article choice errors

The article *a* has been replaced by *the* before proper nouns like *a amazon* and *a himalayas*. The grammar rules are:-

- $X \rightarrow a \text{ himalayas } X_1, \text{ the himalayas } X_1$
- $X \rightarrow a \text{ amazon } X_1, \text{ the amazon } X_1$

The rules suggest that *a himalayas* succeeded by a phrase  $X_1$  can be replaced by *the himalayas* followed by the same phrase.

### 4.2 Preposition errors

Preposition *at* has been replaced by *in* before a place like *at central London*. The grammar rule is:-

- $X \rightarrow X_1 \text{ at central London}, X_1 \text{ in central London}$

### 4.3 Unknown Verb correction

Lets say the training data has these sentences

- $He \text{ like milk} \rightarrow He \text{ likes milk}$
- $They \text{ hate the pollution} \rightarrow They \text{ hate pollution}$

This system will not be able to correct *He hate milk*, because hate needs to be corrected to hates and its grammar has no rule for *hate*  $\rightarrow$  *hates*. But it has a rule for *like*  $\rightarrow$  *likes*. From these two rules, the grammar extractor wont be able to derive *hate*  $\rightarrow$  *hates*. This can be solved by splitting *likes* to *like s*

- $He \text{ like milk} \rightarrow He \text{ like s milk}$

Now extractor will have a rule for this training sentence.

- $X \rightarrow He X_1 \text{ milk}, He X_1 \text{ s milk}$



- $X \rightarrow \text{hate, hate}$

Using these two rules it generates *He hate s milk* from *He hate milk*. Later we combine all the split verbs to get *He hate s milk*.

#### 4.4 Word insertion errors

As the name suggests these errors are due to missing words, e.g.,

- *The court deemed necessary that she respond to the summons.*  $\rightarrow$  *The court deemed it necessary that she respond to the summons.*

For this example the grammar rule extracted is :-

- $X \rightarrow X_1 \text{ deemed } X_2, X_1 \text{ deemed it } X_2$

#### 4.5 Reordering errors

Reordering errors arise due to incorrect ordering of the subject object verb, e.g.,

- Given Hindi sentence:- *सेन्ट्रल लन्डन मे गिरा हेलिकोप्टर*
- Transliteration of Hindi sentence is:- *sentrala landana me giraa helicopters*
- The correct translation of this sentence is:- *helicopter crash in central London*
- Output translation from Hindi-English translation system of this sentence:- *central down in London helicopter.*

If the output translation and correct translation is added to the training corpus of grammar correction system such as,

- *central down in London helicopter*  $\rightarrow$  *helicopter down in central London.*

we can obtain the correct translation.

### 5 Experiments

Now we present the data set and evaluation techniques for our experiment.

#### 5.1 Data set

We ran the grammar correction system on the NUS (NUS Corpus of Learner English) corpus (Dahlmeier et al., 2013). The dataset has been developed at NUS in collaboration with the Centre for English Language Communication (CELC). This is a parallel corpus of 50000 incorrect and correct sentences, all aligned. We took a subset of 4000 line training corpus, 3000 for training and 1000 for testing.

#### 5.2 Cleaning training corpus

This is a preprocessing step before training the grammar correction system. This was primarily due to the presence of noisy data like:-

1. HYPERLINK- <http://en.wikipedia.org/wiki/>
2. Bracketed information:- (DoD) {Common Access Card}
3. Citations:- (Ben, 2008)
4. Presence of sentence pairs without any changes.

### 6 Results

We present the results of SMT based grammar correction in table 1. The results show improvement in BLEU score with increase in the size of training corpus. The baseline is the system which passes incorrect sentences as such i.e., performs *no correction*. We wanted to check what the bleu score would be when no correction is incorporated.

Size of training corpus (sentences)	Size of tuning corpus (sentences)	Size of testing corpus (sentences)	BLEU score
Baseline			0.7551
1000	1000	1000	0.7668
2000	1000	1000	0.7679
3000	1000	1000	0.7744

Table 1: Variation of accuracy with variation of training size

### 7 Conclusion

We have shown how a hierarchical phrase-based MT system like Joshua could be used as a grammar correction system. We observed that increasing training data definitely increases accuracy because patterns in grammar correction keep repeating even if test data is completely different from training set. In future work, we would like to concentrate on “unknown word handling”.

## References

- Chris Brockett, William B. Dolan and Michael Gamon. 2006. *Correcting ESL errors using phrasal SMT techniques*. ACL '06, Sydney, Australia.
- Daniel Dahlmeier, Hwee Tou Ng and Siew Mei Wu. 2013. *Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English*. Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications. BEA 2013, Atlanta, Georgia, USA.
- David Chiang. 2005. *A Hierarchical Phrase-Based Model for Statistical Machine Translation*. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Ann Arbor, Michigan.
- Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi and Hitoshi Isahara. 2003. *Automatic error detection in the Japanese learners' English spoken data*. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 2:145–148. ACL '03, Sapporo, Japan.
- Joel R. Tetreault and Martin Chodorow . 2008. *The ups and downs of preposition error detection in ESL writing*. COLING '08, Manchester, United Kingdom.
- Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post and Chris Callison-Burch. 2012. *Joshua 4.0: packing, PRO, and paraphrases*. Proceedings of the Seventh Workshop on Statistical Machine Translation. WMT '12, Montreal, Canada.
- Kenneth Heafield. 2011. *KenLM: faster and smaller language model queries*. Proceedings of the Sixth Workshop on Statistical Machine Translation. WMT '11, Edinburgh, Scotland.
- Kishore Papineni, Salim Roukos and Todd Ward and Wei-Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Philadelphia, Pennsylvania.
- Martin Chodorow and Claudia Leacock. 2000. *An Unsupervised Method for Detecting Grammatical Errors*. NAACL 2000, Seattle, Washington.
- Matthieu Hermet and Alain Désilets. 2009 *Using first and second language models to correct preposition errors in second language authoring*. EdAppsNLP '09, Boulder, Colorado.
- Omar F. Zaidan. 2009. *Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems*. The Prague Bulletin of Mathematical Linguistics, 91:79–88.
- Percy Liang, Ben Taskar and Dan Klein. . 2006. *Alignment by agreement*. Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. HLT-NAACL '06, New York.
- 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, 1993.
- Rachele De Felice and Stephen G Pulman. 2008. *A classifier-based approach to preposition and determiner error correction in L2 English*. COLING '08, Manchester, United Kingdom.
- Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata and Yuji Matsumoto. 2012. *The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings*. COLING '12, Mumbai, India.
- Y. Albert Park and Roger Levy. 2011. *Automated whole sentence grammar correction using a noisy channel model*. HLT '11, Portland, Oregon.
- Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Ziyuan Wang and Jonathan Weese, and Omar F. Zaidan. 2010. *Joshua 2.0: a toolkit for parsing-based machine translation with syntax, semirings, discriminative training and other goodies*. Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR. WMT '10, Uppsala, Sweden.

# Finding Dependency Parsing Limits over a Large Spanish Corpus

Muntsa Padró<sup>1</sup> Miguel Ballesteros<sup>2</sup> Héctor Martínez<sup>3</sup> Bernd Bohnet<sup>4</sup>

<sup>1</sup>Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

<sup>2</sup>Natural Language Processing Group, Universitat Pompeu Fabra, Barcelona, Spain

<sup>3</sup>Centre for Language Technology, University of Copenhagen, Denmark

<sup>4</sup>School of Computer Science, University of Birmingham, United Kingdom

muntsa.padro@inf.ufrgs.br, miguel.ballesteros@upf.edu  
alonso@hum.ku.dk, bohnetb@cs.bham.ac.uk

## Abstract

This paper studies the performance of different parsers over a large Spanish tree-bank. The aim of this work is to assess the limitations of state-of-the-art parsers. We want to select the most appropriate parser for Subcategorization Frame acquisition, and we focus our analysis on two aspects: the accuracy drop when parsing out-of-domain data, and the performance over specific labels relevant to our task.

## 1 Introduction

Dependency parsing has been addressed from different perspectives, improving performance as better techniques are developed. Nevertheless, we may wonder whether those results are good enough to be useful for tasks that need parsed sentences as input. Depending on the task we want to tackle, we will prefer some labels to be correct with respect to others. For example, in tasks related to extraction of verb complements such as verb Subcategorization Frame (SCF) or Selectional Preference acquisition, we are specially interested in a parser that correctly detects and labels the arguments of the verbs, while we do not need to have a high accuracy in other kind of relations, such as specifiers or modifiers.

In this work, we present a study of the performance of different parsers, following the trend started by McDonald and Nivre (2007) and Hara et al. (2009). We want to maximize the performance of different systems, for Spanish, with the final goal of applying them to concrete tasks, in our case SCF acquisition. For this task, we need to develop a parser that performs well not only in terms of Labeled Attachment Score (LAS), but

also that labels verb complements correctly and that performs well when annotating data that is substantially different from the training corpus.

## 2 Motivation

This work was motivated by the intention of building a SCF acquisition system for Spanish (Padró et al., 2013). SCF acquisition consists of acquiring from data the kind of complements which a verb can appear with (Direct Object, Indirect Object, etc.) and how this complements are fulfilled (Noun Phrase, Clause, etc.). To perform this task, state-of-the-art systems (Briscoe and Carroll, 1997; Korhonen, 2002; Messiant, 2008) use parsed data, where the complements of the verbs are already detected. Thus, the first requirement to develop a SCF acquisition system is to have a parser to annotate the input data.

We started by training *MaltParser* (Nivre and Hall, 2005; Nivre et al., 2007b) for Spanish<sup>1</sup> using IULA Spanish LSP Treebank (Marimon et al., 2012), which is built from technical text. The results obtained in terms of LAS were high but when studying the performance of this parser in terms of which labels and complements were correctly detected, the results showed not to be good enough to lead to satisfactory results in SCF acquisition (Padró et al., 2013). For instance, Indirect Objects (IO) were parsed with F1 around 50%, which means that it will be very unlikely to correctly learn SCFs that contain the very relevant IO label.

Furthermore, we need a parser that performs well when annotating sentences of a domain different from the training Treebank. For that reason we evaluated the parser results over the Tibidabo Treebank (Marimon, 2010), which is made

<sup>1</sup>[http://www.iula.upf.edu/recurs01\\_mpars\\_uk.htm](http://www.iula.upf.edu/recurs01_mpars_uk.htm)

up domain-general texts. Testing on different tests sections that come from different domains is customary for English, where both PTB sections 22 and 23 are used, as well as the Brown corpus. This method is to our knowledge new for Spanish dependency parsing. The results show that, as expected, the performance of the parser over this corpus decreases, making even more difficult the extraction of SCFs.

Thus, we detected two main weaknesses of the parser system: the low performance on labels that may be very important for determined tasks and its dependency on the domain. With that in mind, we evaluated other state-of-the-art parsers (§3.2) to determine which parsers suffer less from these limitations.

### 3 Experiments

#### 3.1 Corpus

We ran our experiments using IULA Spanish LSP Treebank<sup>2</sup> (Marimon et al., 2012). This corpus (henceforth IULA) contains the syntactic annotation of 42,000 sentences (around 590,000 tokens) taken from domain-specific (technical literature) texts. We used the train and test partitions provided by the Treebank developers which are publicly available for replicability.<sup>3</sup>

Furthermore, we used the Tibidabo Treebank (Marimon, 2010) as an alternative test set. Tibidabo contains a set of sentences extracted from the Ancora corpus (Taulé et al., 2008), which was used in the CoNLL-X Shared Task of dependency parsing (Buchholz and Marsi, 2006).

The Tibidabo Treebank was annotated using the same guidelines as IULA Treebank. Therefore, it has the same functions and tag-set as IULA, but since the sentences come from a completely different corpus, it represents a good evaluation frame with regards to the influence of domain change.

In summary, we used a training set to train the different models and two different test sets to evaluate each model. See table 1 for details about the size of the different partitions.

The treebanks used in this work contain up to 25 different dependency relations. In this work we will pay special attention to verbal arguments, i.e., verb complements and subject. Thus, the labels we are interested in are SUBJ (subject), DO (Direct Object), IO (Indirect Object), OBLC (oblique

corpus	sentences	tokens
IULA - Train	33,679	471,624
IULA - Test	8,125	114,610
Tibidabo	3,376	41,620

Table 1: Sizes of the used corpora

complement, a prepositional phrase with bound preposition) and PP-DIR and PP-LOC, which mark prepositional phrases for direction and location respectively.

#### 3.2 Parsers

In what follows we briefly describe the dependency parsers used in our experiments, the parsing approach they belong to, and how we searched for the best possible configuration

##### 3.2.1 Transition-based parser - MaltParser and MaltOptimizer

*MaltParser* (Nivre and Hall, 2005; Nivre et al., 2007b) is a transition-based dependency parser generator. It was one of the best parsers in the CoNLL Shared Tasks in 2006 and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007a) and it contains four different families of transition-based parsers. A transition-based parser is based on an automaton that performs shift-reduce operations, whose transitions manage the input words in order to assign dependencies between them.

*MaltOptimizer* (Ballesteros and Nivre, 2012) is a system designed to optimize MaltParser models by implementing a search of parameters and feature specifications. MaltOptimizer takes a training set in CoNLL data format,<sup>4</sup> and provides an optimal configuration that includes the best parsing algorithm, parsing parameters and a complex feature model over the data structures involved in the parsing process.

MaltOptimizer searches the optimal model maximizing the score of a single evaluation measure, either LAS, LCM (Labeled complete match) or unlabeled evaluation measures. As we mentioned in section 2, our intention is to enhance the performance of specific labels and we have been willing to sacrifice some overall accuracy in favor of better specific models. To this end, we modified the MaltOptimizer source code to make it able to optimize over precision and recall for a specific

<sup>2</sup>[http://iula.upf.edu/recurs01\\_tbk\\_uk.htm](http://iula.upf.edu/recurs01_tbk_uk.htm)

<sup>3</sup><https://repositori.upf.edu/handle/10230/20408>

<sup>4</sup><http://ilk.uvt.nl/conll/#dataformat>

dependency label.<sup>5</sup> Besides improving the accuracy of the parser for a given dependency label, our intention was that we can enhance the general performance of the parser when we optimize over a dependency label which is very frequent. The idea is that the parsers have fewer candidate words for these frequent labels, and therefore, we provide better recall for the rest of labels, thereby reducing error propagation. In our experiments, besides optimizing over general LAS and LCM, we optimized for DO (a very frequent label) and for IO (a rare but relevant label).

In our experiments we ran MaltOptimizer using 5-fold cross-validation over the training corpus in order to ensure the reliability of the outcomes.

### 3.2.2 Maximum Spanning Tree Parser - MSTParser

*MSTParser* is an arc-factored spanning tree parser (McDonald et al., 2005; McDonald et al., 2006). It implements a graph-based second-order parsing model which scores all possible dependency arcs in the sentence and then extracts the dependency tree with the highest score. The score of the trees is calculated basically adding the score of every arc, having at the end a sum with the score of the whole tree.

### 3.2.3 Joint tagger Transition-based parser and Graph-based parser with Hash Kernel and Beam Search- Mate tools

The Mate-Tools provide two parser types: a graph-based (Bohnet, 2010) and a transition-based parser with graph-based re-scoring (completion model) that is able to perform joint PoS tagging and dependency parsing (Bohnet and Kuhn, 2012; Bohnet and Nivre, 2012). We refer to the graph-based parser as Mate-G, to the transition-based parser without graph-based re-scoring as Mate-T, and to the transition-based parser with enabled graph-based completion model to Mate-C. These parsers benefit from a passive-aggressive perceptron algorithm implemented as a Hash Kernel, which makes the parser fast to train and improves accuracy.

Mate-T provides joint PoS tagging and dependency parsing to give account for the interaction between morphology and syntax. It uses a beam search over the space of possible transitions and

<sup>5</sup>The source code and the package with the changes included in MaltOptimizer can be downloaded from: <http://nil.fdi.ucm.es/maltoptimizer/MaltOpt.Specific.zip>

keeps a  $k$  number of possible PoS tags for each word instead of basing its attachment decisions on hard previously calculated PoS tags.

Mate-C is essentially a transition-based parser that uses global information to score again the elements of the beam. The completion model depends on a set of graph parameters called second and third-order factors, which describe the dependency environment of the word, such as the second-order factor that gives account for the head, the dependent and the rightmost grandchild, or the third-order factor that lists the first two children of the dependent.

## 4 Results and Discussion

Table 2 summarizes the results obtained (in terms of LAS) with the different parsers over both test sets. The results obtained over IULA Test are very high, which is probably due to the specificity of the treebank. The results over Tibidabo Treebank can be seen as more general results, and they show how parsers trained with IULA treebank behave when applied to a different domain.

The results given in the table are those obtained with the configuration that leads to better LAS results over IULA Test. The same configuration is used to annotate Tibidabo. For MaltParser, the best LAS was obtained when optimizing the parser for the DO dependency label, which was obtained by applying the MaltOptimizer modifications that we explained in section 3.2.1. This therefore confirms our expectations that optimizing over very frequent labels may improve the overall accuracy<sup>6</sup>. The algorithm selected by MaltOptimizer was Covington non-projective, which is described by Covington (2001) and included in MaltParser by Nivre (2008). For the Mate parsers, we used the default training settings for the graph-based parser. For the transition-based parser, we used a beam size of 40 and 25 training iterations.

Parser	IULA Test LAS (%)	Tibidabo LAS (%)
Malt	93.16	89.04
MST	92.72	89.36
Mate-T	94.47	91.05
Mate-C	94.70	91.43
Mate-G	94.49	91.26

Table 2: Obtained LAS for each parser

<sup>6</sup>Optimizing over IO did not improve significantly the results over that complement nor overall LAS

In the table, we can see that the differences of performance assert the domain difference between the corpora and that Mate parsers clearly outperform the others. The best performance is obtained with Mate-C<sup>7</sup>.

It is not surprising that the best results are obtained with the Mate parsers since those parsers use enhanced parsing models. Nevertheless, to obtain these high results it was necessary to change the treebank configuration to use the short PoS (this is, just the category) in the position of the long PoS, and keep the long PoS (i.e, the morphology) in the feature column. When using the original configuration of the treebank (long PoS) the results obtained were much lower, and specially suffered from the domain change (LAS=93.69% for IULA Test and LAS=88.77% for Tibidabo). The Mate parsers were optimized for for the usage of CoNLL-09 data format which includes PoS tags and morphological features only. Under these conditions, the short PoS tags fit best into the PoS column and the fine grained tags into the morphologic column. The other parsers use CoNLL-X format. MST uses all columns for training, and Malt only uses the features provided in the feature model which is one of the outputs of MaltOptimizer, but changing the data did not improve the results, neither for Malt nor for MST.

#### 4.1 Specific Label Performance

The results obtained in terms of LAS are very satisfactory (the best parsing results reported for Spanish so far), specially for Mate parsers. Nevertheless, when we study the performance over concrete labels, we see that we can not rely on the parser for some of them. Table 3 presents the results for the labels we are interested in (§3.1). The table shows Precision, Recall and F1 scores obtained with Mate-C, which is the parser that performs better not only in terms LAS but also for individual complements. The table also shows the relative frequency of the complements in each corpus. Note that some of the studied complements are terribly infrequent.

Note that, from the labels we are interested in, just the frequent ones (SUBJ and DO) are annotated with high F1. OBLC has acceptable results, but the other complements are a big source of error, having low P and R. One of the goals of the

<sup>7</sup>All differences are significant (using T-test with  $\alpha$  set to 0.05) except between Mate-C and Mate-G over Tibidabo

Label	IULA Test				Tibidabo			
	Freq.	P	R	F1	Freq.	P	R	F1
SUBJ	5.90	93.23	93.43	93.33	7.35	89.12	88.66	88.89
DO	4.64	93.02	93.25	93.13	7.03	85.84	85.64	85.74
IO	0.09	67.90	51.89	58.83	0.46	66.67	48.42	56.10
OBLC	0.20	83.56	83.49	83.53	1.30	75.25	69.00	71.99
PP-DIR	0.05	56.67	43.59	49.28	0.18	57.14	16.44	25.53
PP-LOC	0.03	61.84	39.83	48.45	0.14	56.00	24.56	34.14

Table 3: Results for some labels with Mate-C. All figures are percentages.

present work was to see whether it was possible to build a parser that had better performance for the critical complements (specially in terms of Precision) even if it had worse LAS. Nevertheless, the results showed that even with the parser that performed better, the Precision and Recall of the infrequent complements is too low to obtain good results in subsequent tasks that require high label-specific performance, as shown by Padró et al. (2013) for SCF acquisition.

## 5 Conclusions and Future Work

This work studied the limitations of state-of-the-art parsers. We trained different systems over a large Spanish treebank and tested them over a treebank from a different domain. Our experiments show that though the obtained LAS is high, the performance over some concrete labels is very low in all cases, limiting the usability of the parsed data for tasks than rely on label-specific accuracy.

One important future line is to look for parser modifications that allow the system to perform better in the labels we are interested in. To do so, one idea would be to use semantic features to give more information to the parser like in (Agirre et al., 2012). We did some preliminary experiments in that line, using information about the semantic classes for common nouns (specifically for the classes human and location), but the results showed that this information did not lead to a better performance of the parser. This is probably due to the sparsity of this information, but it is still an interesting line to study, since it lead to good results in other cases (Agirre et al., 2012).

## Acknowledgements

We thank Núria Bel and Joakim Nivre for their useful comments. HM has been partially funded by the European Commission’s 7th Framework Program under grant agreement 238405 (CLARA). MP has been funded by the European Project PANACEA (FP7-ICT-2010- 248064).

## References

- Eneko Agirre, Aitziber Atutxa, and Kepa Sarasola. 2012. Contribution of complex lexical information to solve syntactic ambiguity in basque. In *Conference on Computational Linguistics, COLING 2012*, Mumbai, India, 12/2012.
- Miguel Ballesteros and Joakim Nivre. 2012. Malt-Optimizer: A System for MaltParser Optimization. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*.
- Bernd Bohnet and Jonas Kuhn. 2012. The best of both worlds - a graph-based completion model for transition-based parsers. In *EACL*.
- Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*, pages 89–97.
- Ted Briscoe and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *ANLP*, pages 356–363.
- Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164.
- Michael A. Covington. 2001. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102.
- Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. 2009. Effective analysis of causes and interdependencies of parsing errors. In *Proceedings of the 11th International Conference on Parsing Technologies, IWPT '09*, pages 180–191, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Anna Korhonen. 2002. *Subcategorization acquisition*. Ph.D. thesis, February.
- M. Marimon, B. Fisas, N. Bel, B. Arias, S. Vázquez, J. Vivaldi, S. Torner, M. Villegas, and M. Lorente. 2012. The iula treebank.
- Montserrat Marimon. 2010. The tibidabo treebank. *Procesamiento del lenguaje natural*, 2010, vol. 45, num. 1, p. 113-119.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530.
- Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 216–220.
- Cédric Messiant. 2008. A subcategorization acquisition system for french verbs. In *ACL (Student Research Workshop)*, pages 55–60.
- Joakim Nivre and Johan Hall. 2005. MaltParser: A language-independent system for data-driven dependency parsing. In *Proceedings of the 4th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 137–148.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007a. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 915–932.
- Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chaney, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007b. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553.
- Muntsa Padró, Núria Bel, and Aina Garí. 2013. Verb SCF extraction for Spanish with dependency parsing. *Procesamiento del Lenguaje Natural*, (51), September.
- Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA).

# High Quality Dependency Selection from Automatic Parses

Gongye Jin, Daisuke Kawahara, Sadao Kurohashi

Graduate School of Informatics, Kyoto University

Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501, Japan

jin@nlp.ist.i.kyoto-u.ac.jp {dk, kuro}@i.kyoto-u.ac.jp

## Abstract

Many NLP tasks such as question answering and knowledge acquisition are tightly dependent on dependency parsing. Dependency parsing accuracy is always decisive for the performance of subsequent tasks. Therefore, reducing dependency parsing errors or selecting high quality dependencies is a primary issue. In this paper, we present a supervised approach for automatically selecting high quality dependencies from automatic parses. Experimental results on three different languages show that our approach can effectively select high quality dependencies from the result analyzed by a dependency parser.

## 1 Introduction

Knowledge acquisition from a large corpus has been actively studied recently. Knowledge is often acquired from the fundamental analysis. In particular, dependency parsing has been used for some tasks like case frame compilation (Kawahara and Kurohashi, 2006), relation extraction (Saeger et al., 2011) and paraphrase acquisition (Hashimoto et al., 2011). For these tasks, the accuracy of dependency parsing is vital. Although the accuracy of state-of-the-art dependency parsers for some languages like English or Japanese is over 90%, it is still not high enough to acquire accurate knowledge, not to mention those difficult-to-analyze languages like Chinese and Arabic.

Instead of using all the automatic parses, it is possible to use only high quality dependencies for knowledge acquisition. In this paper, we present a supervised approach for selecting high quality dependencies from automatic dependency parses. This method considers linguistic features that are related to the difficulty of dependency parsing. The experimental results on English, Chinese and

Japanese show that our proposed method can select dependencies of higher quality than baseline methods for all the languages.

## 2 Related Work

There have been a few approaches devoted to automatic selection of high quality parses or dependencies. According to selection algorithms, they can be categorized into supervised and unsupervised.

Supervised methods mainly focus on the construction of a machine learning classifier to predict the reliability of parses or dependencies based on various kinds of features both on syntactic and semantic level. Yates et al. (2006) created WOODWARD which is a Web-based semantic filtering system. Kawahara and Uchimoto (2008) built a binary classifier that classifies each parse of a sentence as reliable or not. Among supervised methods, ensemble approaches were also proposed. Reichart and Rappoport (2007) detected parse quality by a Sample Ensemble Parse Assessment (SEPA) algorithm. Another similar approach proposed by Sagae and Tsujii (2007) also selected high quality parses by computing the level of agreement on different parser outputs. Iwatate (2012) applied a tournament model on Japanese dependency parsing and then selected reliable dependencies by using SVM output. The work most related to ours is the work of Yu et al. (2008). They proposed a framework that selects high quality parses in the first stage, and then selected high quality dependencies from the filtered parses. In comparison, we consider that even some low quality sentences possibly contain high quality dependencies. Also, we take into account other aspects that can affect high quality dependency classification and create a new set of linguistic features for classification.

Also, unsupervised algorithms for detecting reliable dependency parses were proposed. Reichart



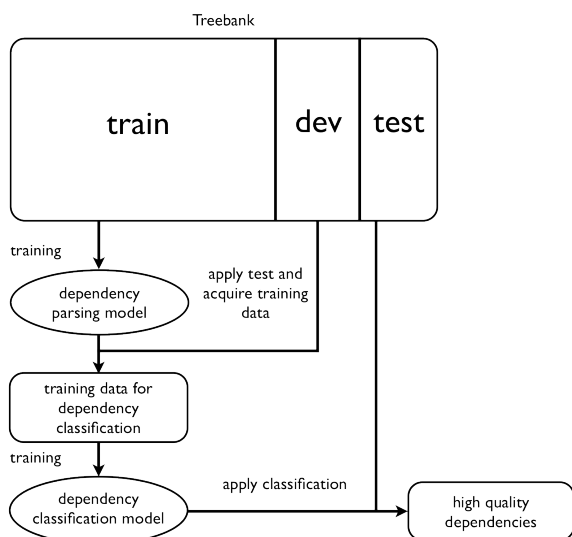


Figure 1: Overview of high quality dependency selection

and Rappoport (2009) proposed an unsupervised method for high quality parse selection, which was based on the idea that syntactic structures that are frequently created by a parser are more likely to be correct. Dell’Orletta et al. (2011) proposed ULISSE (Unsupervised LINGuisticallydriven Selection of dEpendency parses), which uses an unsupervised method in a supervised parsing scenario. Although unsupervised methods may solve the domain adaption issue and do not use any annotated data, the accuracy of selected parses, which is under 95%, still needs to be improved for knowledge acquisition tasks.

### 3 High Quality Dependency Selection

In this section, we present a framework of highly reliable dependency selection from automatic parses. Figure 1 shows the overview of our approach. We use a part of a treebank to train a parser and another part to train a binary classifier which judges a dependency to be reliable or not.

#### 3.1 Training Data for Dependency Classification

We collect training data from the same corpus which is also used in dependency parsing in the first stage. First, the training section is used to train a dependency parser and the development section is used to apply dependency parsing using the model trained by training section. From the parses of the development section, we acquire training data for dependency classification

by labeling each dependency according to the gold standard data. All the correct dependencies are defined as reliable and vice versa.

#### 3.2 Features for Dependency Classification

Most basic features consider that word pairs are much less likely to have a dependency relation when there are punctuation between them. On the other hand, based on the fact that dependencies with longer distance always show worse parsing performance (McDonald and Nivre, 2007), distance is another important factor that reflects the difficulty of judging whether two words have a dependency relation. Yu et al. (2008) used the features mentioned above and PoS features except the word features and did not use the context features, which are described later.

In addition to these basic features, we consider context features that are thought to affect the parsing performance. Table 2 lists these context features. In some more complex cases, it is also necessary to observe larger span of context. In order to learn such linguistic characteristics automatically, besides POS tags the head and modifier in a dependency, we also use their preceding and following one and two words along with their POS tags.

Another important fact is that verbal phrases in the dependency tree structure of a parse are normally the root node of the whole dependency tree or the parent node of a subtree. When a word pair that contains a verbal phrase between them, the two words are always on different sides of a parent node. Thus, these kinds of word pairs will always have no dependency link between them. This leads to the fact that argument pairs that have a verb between them rarely have a dependency relation. Observing whether there are verbal phrases between a head-modifier pairs can help judge whether the dependency between them is reliable.

The input of our high quality dependency selection method is a dependency tree. It is very natural to use tree-based features to identify the quality of dependencies. Based on a head-modifier dependency pair, we observe modifier’s modifiers, i.e. children nodes. We use the leftmost and rightmost of children nodes to represent all the children nodes. We also take head’s parent node into consideration, which we call a modifier’s grandparent node. Furthermore, children nodes of the

grandparent node which we call a modifier’s uncle nodes are also considered as other features. Similarly, we use leftmost and rightmost uncle nodes.

## 4 Experiments

### 4.1 Experimental Settings

We first experiment on English, Chinese and Japanese. For English, we employ MSTparser<sup>1</sup> as a base dependency parser and use sections 02 to 21 from Wall Street Journal (WSJ) corpus in Penn Treebank (PTB) to train a dependency parsing model. Then, we use section 22 from WSJ to apply the dependency parsing model to acquire the training data for dependency classification. MXPOST<sup>2</sup> tagger is used for English automatic POS tagging. For Chinese, we use CNP (Chen et al., 2009) parser to train a dependency parser using section 1 to 270, 400 to 931 and 1001 to 1151 from Penn Chinese Treebank (CTB). Sections 301 to 325 are used to apply dependency parsing to acquire training data for dependency classification. We use MMA (Kruengkrai et al., 2009) to apply both segmentation and POS tagging. Different from the previous two languages which take *words* as the basic unit, experiments on Japanese are based on the unit of the phrase segments *bunsetsu*. We first use JUMAN<sup>3</sup> for Japanese morphological analysis. Then KNP<sup>4</sup> is utilized for Japanese dependency parsing. Section 950112, 950113 and 9509ED from Kyoto Corpus are used to apply dependency parsing and acquire training data for dependency selection.

We employ SVM-Light<sup>5</sup> with polynomial kernel (degree 3) to solve the binary classification. In order to compare with previous work by Yu et al. (2008), we use the basic feature set as a baseline. For English, section 23 from WSJ is used as a test set. Section 271 to 300 from CTB, and section 950114 to 950117 and 9510ED to 9512ED from Kyoto Corpus are used to test the classification approach in Chinese and Japanese, respectively. are used to test the classification approach in Chinese and Japanese respectively.

<sup>1</sup><http://sourceforge.net/projects/mstparser/>

<sup>2</sup>[http://www.inf.ed.ac.uk/resources/nlp/local\\_doc/MXPOST.html](http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html)

<sup>3</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

<sup>4</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP>

<sup>5</sup><http://svmlight.joachims.org>

According to the output of the SVM, we only select dependencies that have the score higher than a threshold. Precision is calculated as the ratio of correct dependencies in retrieved ones. Recall is the ratio of correct dependencies in total. In Chinese and Japanese, we treat incorrect segmentations as incorrect dependencies. Note that the maximum recall value equals the precision of base dependency parser without dependency selection.

### 4.2 Experimental Results

#### 4.2.1 Effectiveness of Dependency Selection

Figure 2 shows the precision-recall curves of the classification using SVM for three languages. In these graphs, ‘basic’ means the method using the basic features, ‘context’ stands for the method with context information, and ‘context+tree’ means the method with additional tree-based features.

One of the biggest problems that most data-driven parsers are facing is the domain adaption problem. When they are applied to a text of a different domain, their accuracy decreases significantly. We applied the dependency parsing model trained on WSJ to the Brown corpus, and obtained an unlabeled attachment score of 0.832, which is significantly lower than the in-domain score by 8.1%. We applied the same dependency selection model trained on WSJ to the Brown corpus. Figure 4 shows the precision-recall curves of dependency selection on the Brown corpus. From the results, we can see that when the recall is 40% for example, high quality dependencies with a precision of over 95% can be acquired. This shows that our method works well on data from different domains. This fact creates a good way to acquire knowledge from a large raw corpus in different domains (e.g., the Web).

#### 4.2.2 Statistics of Selected Dependencies

In this section, in order to know what kind of dependencies are mainly selected, we show an investigation on the distribution of types of dependencies. Each dependency type in English and Chinese is represented by the coarse-grained POS pairs (the first two characters of POS names). Japanese dependencies are represented by the translated POS tags of *Bunsetsu* pairs. Figure 3 shows the statistics of POS pairs in three different languages. the leftmost graphs are drawn without selection. The middle and right graphs stand for

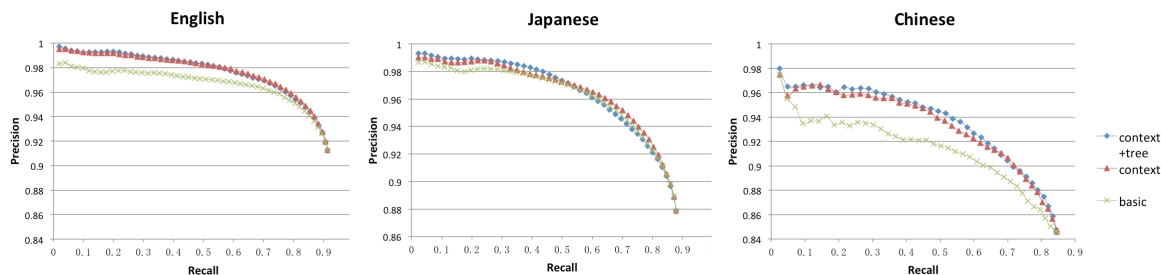


Figure 2: Precision-recall curves of dependency classification for English (left), Japanese (middle) and Chinese (right)

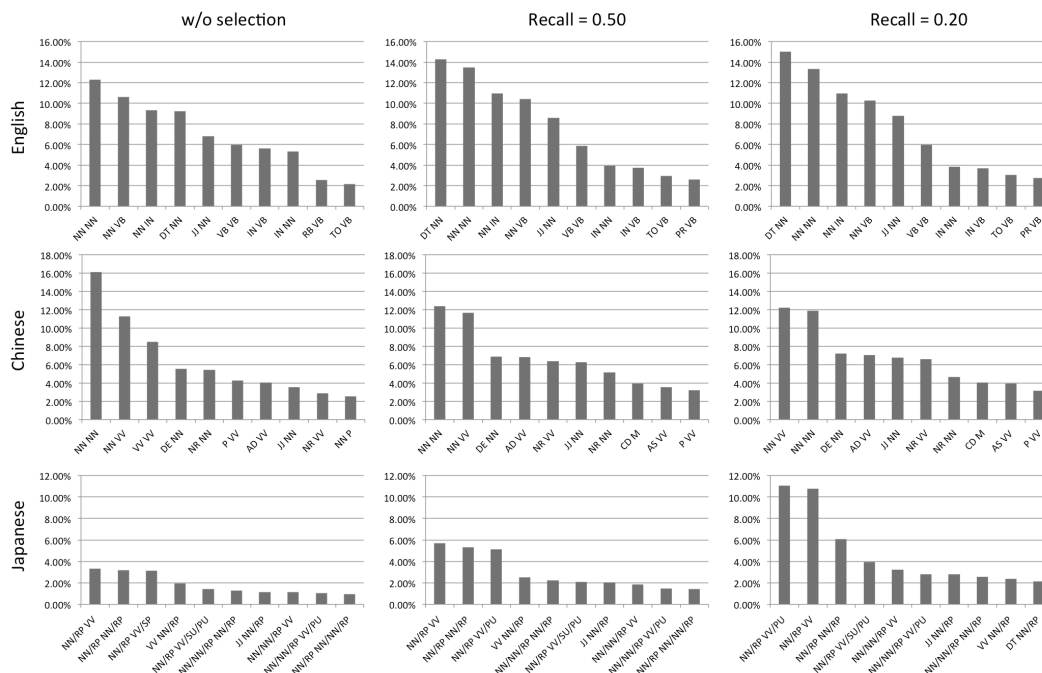


Figure 3: Statistics of POS tags of dependencies in different languages: dependencies without selection (left), dependencies when recall is 50% (middle), dependencies when recall is 20% (right)

the dependencies selected under different thresholds (i.e., recall is 20% and recall is 50% respectively). We found that dependencies with nouns are dominant in all the types for all the languages. Secondly, dependencies related to verbs which are very informative patterns account for a large proportion.

## 5 Conclusion and Future Work

In this paper, we proposed a classification approach for high quality dependency selection. We created new sets of features to select highly reliable dependencies from each parse through a parser. The experiments showed that our method worked for in-domain parses and also out-of-domain parses. We can extract high quality dependencies from a large corpus such as the Web and subsequently assist knowledge acquisition tasks,

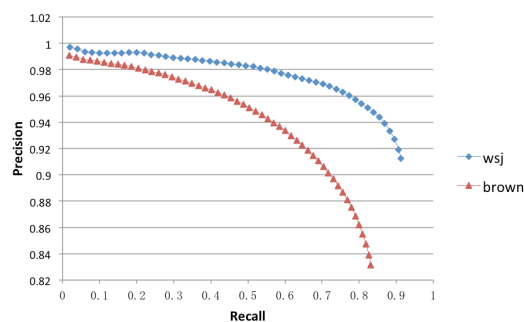


Figure 4: Precision-recall curves of dependency selection on Brown corpus

such as subcategorization frame acquisition and case frame compilation (Kawahara and Kurohashi, 2010), which depends highly on the parse quality. We also plan to use a bootstrapping strategy to improve a dependency parser based on acquired high quality knowledge from large corpora.

## References

- Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimota, and Kentaro Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of EMNLP 2009*, pages 570–579.
- Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2011. Ulisse: an unsupervised algorithm for detecting reliable dependency parses. In *Proceeding of CoNLL 2011*, pages 115–124.
- Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jun'ichi Kazama, and Sadao Kurohashi. 2011. Extracting paraphrases from definition sentences on the web. In *Proceedings of ACL 2011*, pages 1087–1097.
- Masakazu Iwatate. 2012. *Development of Pairwise Comparison-based Japanese Dependency Parsers and Application to Corpus Annotation*. Ph.D. thesis, NAIST, Japan.
- Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of HLT-NAACL 2006*, pages 176–183.
- Daisuke Kawahara and Sadao Kurohashi. 2010. Acquiring reliable predicate-argument structures from raw corpora for case frame compilation. In *Proceedings of LREC 2010*, pages 1389–1393.
- Daisuke Kawahara and Kiyokata Uchimoto. 2008. Learning reliability of parses for domain adaptation. In *Proceedings of IJCNLP 2008*, pages 709–714.
- Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In *Proceedings of ACL-IJCNLP 2009*, pages 513–521.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL 2007*, pages 122–131.
- Roi Reichart and Ari Rappoport. 2007. An ensemble method for selection of high quality parses. In *Proceedings of ACL 2007*, pages 408–415.
- Roi Reichart and Ari Rappoport. 2009. Automatic selection of high quality parses created by a fully unsupervised parser. In *Proceedings of CoNLL 2009*, pages 156–164.
- Stijn De Saeger, Kentaro Torisawa, Masaaki Tsuchida, Jun'ichi Kazama, Chikara Hashimoto, Ichiro Yamada, Jong Hoon Oh, István Varga, and Yulan Yan. 2011. Relation acquisition using word classes and partial patterns. In *Proceedings of EMNLP 2011*, pages 825–835.
- Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensemble. In *Proceedings of EMNLP-CoNLL 2007*, pages 408–415.
- Alexander Yates, Stefan Schoenmackers, and Oren Etzioni. 2006. Detecting parser errors using web-based semantic filters. In *Proceedings of EMNLP 2006*, pages 27–34.
- Kun Yu, Daisuke Kawahara, and Sadao Kurohashi. 2008. Cascaded classification for high quality head-modifier pair selection. In *Proceedings of NLP 2008*, pages 1–8.

# Building Specialized Bilingual Lexicons Using Word Sense Disambiguation

**Dhouha Bouamor**  
CEA, LIST, Vision and  
Content Engineering Laboratory,  
91191 Gif-sur-Yvette CEDEX  
France  
dhouha.bouamor@cea.fr

**Nasredine Semmar**  
CEA, LIST, Vision and Content  
Engineering Laboratory,  
91191 Gif-sur-Yvette  
CEDEX France  
nasredine.semmar@cea.fr

**Pierre Zweigenbaum**  
LIMSI-CNRS,  
F-91403 Orsay CEDEX  
France  
pz@limsi.fr

## Abstract

This paper presents an extension of the standard approach used for bilingual lexicon extraction from comparable corpora. We study the ambiguity problem revealed by the seed bilingual dictionary used to translate context vectors and augment the standard approach by a Word Sense Disambiguation process. Our aim is to identify the translations of words that are more likely to give the best representation of words in the target language. On two specialized French-English and Romanian-English comparable corpora, empirical experimental results show that the proposed method consistently outperforms the standard approach.

## 1 Introduction

Over the years, bilingual lexicon extraction from comparable corpora has attracted a wealth of research works (Fung, 1998; Rapp, 1995; Chiao and Zweigenbaum, 2003). The main work in this research area could be seen as an extension of Harris's *distributional hypothesis* (Harris, 1954). It is based on the simple observation that a word and its translation are likely to appear in similar contexts across languages (Rapp, 1995). Based on this assumption, the alignment method, known as the *standard approach* builds and compares context vectors for each word of the source and target languages.

A particularity of this approach is that, to enable the comparison of context vectors, it requires the existence of a seed bilingual dictionary to translate source context vectors. The use of the bilingual dictionary is problematic when a word has several translations, whether they are synonymous or

polysemous. For instance, the French word *action* can be translated into English as *share*, *stock*, *lawsuit* or *deed*. In such cases, it is difficult to identify in flat resources like bilingual dictionaries, wherein entries are usually unweighted and unordered, which translations are most relevant. The standard approach considers all available translations and gives them the same importance in the resulting translated context vectors independently of the domain of interest and word ambiguity. Thus, in the financial domain, translating *action* into *deed* or *lawsuit* would probably introduce noise in context vectors.

In this paper, we present a novel approach which addresses the word ambiguity problem neglected in the standard approach. We introduce a use of a WordNet-based semantic similarity measure permitting the disambiguation of translated context vectors. The basic intuition behind this method is that instead of taking all translations of each seed word to translate a context vector, we only use the translations that are more likely to give the best representation of the context vector in the target language. We test the method on two comparable corpora specialized on the Breast Cancer domain, for the French-English and Romanian-English pair of languages. This choice allows us to study the behavior of the disambiguation for a pair of languages that are richly represented and for a pair that includes Romanian, a language that has fewer associated resources than French and English.

## 2 Related Work

Recent improvements of the standard approach are based on the assumption that the more the context vectors are representative, the better the bilingual lexicon extraction is. Prochasson et al. (2009)

used transliterated words and scientific compound words as ‘anchor points’. Giving these words higher priority when comparing target vectors improved bilingual lexicon extraction. In addition to transliteration, Rubino and Linarès (2011) combined the contextual representation within a thematic one. The basic intuition of their work is that a term and its translation share thematic similarities. Hazem and Morin (2012) recently proposed a method that filters the entries of the bilingual dictionary based upon POS-tagging and domain relevance criteria, but no improvements was demonstrated.

Gaussier et al. (2004) attempted to solve the problem of different word ambiguities in the source and target languages. They investigated a number of techniques including canonical correlation analysis and multilingual probabilistic latent semantic analysis. The best results, with a very small improvement were reported for a mixed method. One important difference with Gaussier et al. (2004) is that they focus on words ambiguities on source and target languages, whereas we consider that it is sufficient to disambiguate only translated source context vectors.

### 3 Context Vector Disambiguation

The approach we propose augments the standard approach used for bilingual lexicons mining from comparable corpora. As it was mentioned in section 1, when the lexical extraction applies to a specific domain, not all translations in the bilingual dictionary are relevant for the target context vector representation. For this reason, we introduce a WordNet-based WSD process that aims at improving the adequacy of context vectors and therefore improve the results of the standard approach.

A large number of WSD techniques were previously proposed in the literature. The most popular ones are those that compute semantic similarity with the help of existing thesauri such as WordNet (Fellbaum, 1998). This thesaurus has been applied to many tasks relying on word-based similarity, including document (Hwang et al., 2011) and image (Cho et al., 2007; Choi et al., 2012) retrieval systems. In this work, we use this resource to derive a semantic similarity between lexical units within the same context vector. To the best of our knowledge, this is the first application of WordNet to the task of bilingual lexicon extraction from comparable corpora.

Once translated into the target language, the context vectors disambiguation process intervenes. This process operates *locally* on each context vector and aims at finding the most prominent translations of polysemous words. For this purpose, we use monosemic words as a seed set of disambiguated words to infer the polysemous word’s translations senses. We hypothesize that a word is monosemic if it is associated to only one entry in the bilingual dictionary. We checked this assumption by probing monosemic entries of the bilingual dictionary against WordNet and found that 95% of the entries are monosemic in both resources.

Formally, we derive a semantic similarity value between all the translations provided for each polysemous word by the bilingual dictionary and all monosemic words appearing within the same context vector. There is a relatively large number of word-to-word similarity metrics that were previously proposed in the literature, ranging from path-length measures computed on semantic networks, to metrics based on models of distributional similarity learned from large text collections. For simplicity, we use in this work, the Wu and Palmer (1994) (WUP) path-length-based semantic similarity measure. It was demonstrated by (Lin, 1998) that this metric achieves good performances among other measures. WUP computes a score (equation 1) denoting how similar two word senses are, based on the depth of the two synsets ( $s_1$  and  $s_2$ ) in the WordNet taxonomy and that of their Least Common Subsumer ( $LCS$ ), i.e., the most specific word that they share as an ancestor.

$$WupSim(s_1, s_2) = \frac{2 \times depth(LCS)}{depth(s_1) + depth(s_2)} \quad (1)$$

In practice, since a word can belong to more than one synset in WordNet, we determine the semantic similarity between two words  $w_1$  and  $w_2$  as the maximum  $WupSim$  between the synset or the synsets that include the  $synsets(w_1)$  and  $synsets(w_2)$  according to the following equation:

$$SemSim(w_1, w_2) = \max\{WupSim(s_1, s_2); (s_1, s_2) \in synsets(w_1) \times synsets(w_2)\} \quad (2)$$

Then, to identify the most prominent translations of each polysemous unit  $w_p$ , an *average similarity* is computed for each translation  $w_p^j$  of  $w_p$ :

$$AveSim(w_p^j) = \frac{\sum_{i=1}^N SemSim(w_i, w_p^j)}{N} \quad (3)$$

Corpus	French	English
	396,524	524,805
Corpus	Romanian	English
	22,539	322,507

Table 1: Comparable corpora sizes in term of words.

where  $N$  is the total number of monosemic words and  $Sem_{Sim}$  is the similarity value of  $w_p^j$  and the  $i^{th}$  monosemic word. Hence, according to average relatedness values  $Ave\_Sim(w_p^j)$ , we obtain for each polysemous word  $w_p$  an ordered list of translations  $w_p^1 \dots w_p^n$ . This allows us to select translations of words which are more salient than the others to represent the word to be translated.

## 4 Experiments and Results

### 4.1 Resources

#### 4.1.1 Comparable corpora

We conducted our experiments on two French-English and Romanian-English comparable corpora specialized on the *breast cancer* domain. Both corpora were extracted from Wikipedia<sup>1</sup>. We consider the topic in the source language (for instance *cancer du sein* [breast cancer]) as a query to Wikipedia and extract all its sub-topics (i.e., sub-categories in Wikipedia) to construct a domain-specific *category tree*. Then, based on the constructed tree, we collect all Wikipedia pages belonging to one of these categories and use *inter-language links* to build the comparable corpus. Both corpora were normalized through the following linguistic preprocessing steps: tokenisation, part-of-speech tagging, lemmatisation, and function word removal. The resulting corpora<sup>2</sup> sizes are given in Table 1.

#### 4.1.2 Bilingual dictionary

The French-English bilingual dictionary used to translate context vectors consists of an in-house manually revised bilingual dictionary which contains about 120,000 entries belonging to the general domain. It is important to note that words has on average 7 translations in the bilingual dictionary. The Romanian-English dictionary consists of translation pairs extracted from Wikipedia.

<sup>1</sup><http://dumps.wikimedia.org/>

<sup>2</sup>Comparable corpora will be shared publicly

The resulting bilingual dictionary contains about 136,681 entries for Romanian-English with an average of 1 translation per word.

#### 4.1.3 Evaluation list

In bilingual terminology extraction from comparable corpora, a reference list is required to evaluate the performance of the alignment. Such lists are usually composed of about 100 single terms (Hazem and Morin, 2012; Chiao and Zweigenbaum, 2002). Here, we created a reference list<sup>3</sup> for each pair of language. The French-English list contains 96 terms extracted from the French-English MESH and the UMLS thesauri<sup>4</sup>. The Romanian-English reference list was created by a native speaker and contains 38 pair of words. Note that reference terms pairs appear at least five times in each part of both comparable corpora.

### 4.2 Experimental setup

Three other parameters need to be set up: (1) the window size, (2) the association measure and the (3) similarity measure. To define context vectors, we use a seven-word window as it approximates syntactic dependencies. Concerning the rest of the parameters, we followed Laroche and Langlais (2010) for their definition. The authors carried out a complete study of the influence of these parameters on the bilingual alignment and showed that the most effective configuration is to combine the Discounted Log-Odds ratio (equation 4) with the cosine similarity. The Discounted Log-Odds ratio is defined as follows:

$$Odds-Ratio_{disc} = \log \frac{(O_{11} + \frac{1}{2})(O_{22} + \frac{1}{2})}{(O_{12} + \frac{1}{2})(O_{21} + \frac{1}{2})} \quad (4)$$

where  $O_{ij}$  are the cells of the  $2 \times 2$  contingency matrix of a token  $s$  co-occurring with the term  $S$  within a given window size.

### 4.3 Results and discussion

It is difficult to compare results between different studies published on bilingual lexicon extraction from comparable corpora, because of difference between (1) used corpora (in particular their construction constraints and volume), (2) target domains, and also (3) the coverage and relevance of linguistic resources used for translation. To the best of our knowledge, there is no common benchmark that can serve as a reference. For this reason,

<sup>3</sup>Reference lists will be shared publicly

<sup>4</sup><http://www.nlm.nih.gov/>

		Method	WN-T <sub>1</sub>	WN-T <sub>2</sub>	WN-T <sub>3</sub>	WN-T <sub>4</sub>	WN-T <sub>5</sub>	WN-T <sub>6</sub>	WN-T <sub>7</sub>
b) FR-EN		Standard Approach(SA)	0.49						
	Single measures	WUP	0.48	0.56	0.56	0.54	0.55	0.54	0.55
		PATH	0.54	0.54	0.55	0.56	0.57	0.55	0.55
		LEACOCK	0.50	0.57	0.55	0.56	0.54	0.55	0.54
		LESK	0.46	0.54	0.54	<b>0.59</b>	0.55	0.55	0.54
		VECTOR	0.51	0.56	0.53	0.56	0.54	0.56	0.55
		Method	WN-T <sub>1</sub>	WN-T <sub>2</sub>	WN-T <sub>3</sub>	WN-T <sub>4</sub>	WN-T <sub>5</sub>	WN-T <sub>6</sub>	WN-T <sub>7</sub>
b) RO-EN		Standard Approach(SA)	0.21						
	Single measures	WUP	0.18	0.21	0.21	0.21	0.21	0.21	0.21
		PATH	0.18	0.21	0.21	0.21	0.21	0.21	0.21
		LEACOCK	0.15	0.18	0.18	0.18	0.18	0.18	0.18
		LESK	0.21	0.21	0.21	0.21	0.21	0.21	0.21
		VECTOR	0.18	0.21	0.21	0.21	0.21	0.21	0.21

Table 2: F-Measure at Top20 for the Breast Cancer domain for the two pairs of languages; In each column, italics shows best single similarity measure, bold shows best result. Underline shows best result overall.

we use the results of the standard approach (SA) as a reference. We evaluate the performance of both the SA and ours with respect to Top20 F-Measure which computes the harmonic mean between precision and recall.

Our method provides a ranked list of translations for each polysemous word. A question that arises here is whether we should introduce only the best ranked translation in the context vector or consider a larger number of words, especially when a translations list contain synonyms. For this reason, we take into account in our experiments different number of translations, noted WN-T<sub>*i*</sub>, ranging from the pivot translation ( $i = 1$ ) to the seventh word in the translations list. This choice is motivated by the fact that words in the French-English corpus have on average 7 translations in the bilingual dictionary. The baseline (SA) uses all translations associated to each entry in the bilingual dictionary. Table 2a displays the results obtained for the French-English comparable corpus. The first substantial observation is that our method which consists in disambiguating polysemous words within context vectors consistently outperforms the standard approach. The maximum F-measure was obtained by LESK when for each polysemous word up to four translations (WN-T<sub>4</sub>) are considered in context vectors. This method achieves an improvement of +10% and over the standard approach.

Concerning the Romanian-English pair of lan-

guage, no improvements have been reported. The reason being that words in the bilingual dictionary are not heavily polysemous. Each word used to shape context vectors is associated to only one translation in the bilingual dictionary.

## 5 Conclusion

We presented in this paper a novel method that extends the standard approach used for bilingual lexicon extraction from comparable corpora. The proposed method disambiguates polysemous words in context vectors and selects only the translations that are most relevant to the general context of the corpus. Conducted experiments on a highly polysemous specialized comparable corpus show that integrating such process leads to a better performance than the standard approach. Although our initial experiments are positive, we believe that they could be improved in a number of ways. It would also be interesting to mine much more larger comparable corpora and focus on their quality as presented in (Li and Gaussier, 2010). We want also to test our method on bilingual lexicon extraction for a larger panel of specialized corpora, where disambiguation methods are needed to prune translations that are irrelevant to the domain.

## References

Yun-Chuang Chiao and Pierre Zweigenbaum. 2002. Looking for candidate translational equivalents in



- specialized, comparable corpora. In *Proceedings of the 19th international conference on Computational linguistics - Volume 2, COLING '02*, pages 1–5. Association for Computational Linguistics.
- Yun-Chuang Chiao and Pierre Zweigenbaum. 2003. The effect of a general lexicon in corpus-based identification of french-english medical word translations. In *Proceedings Medical Informatics Europe, volume 95 of Studies in Health Technology and Informatics*, pages 397–402, Amsterdam.
- Miyoung Cho, Chang Choi, Hanil Kim, Jungpil Shin, and PanKoo Kim. 2007. Efficient image retrieval using conceptualization of annotated images. *Lecture Notes in Computer Science*, pages 426–433. Springer.
- Dongjin Choi, Jungin Kim, Hayoung Kim, Myungwon Hwang, and Pankoo Kim. 2012. A method for enhancing image retrieval based on annotation using modified wup similarity in wordnet. In *Proceedings of the 11th WSEAS international conference on Artificial Intelligence, Knowledge Engineering and Data Bases, AIKED'12*, pages 83–87, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Pascale Fung. 1998. A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In *Parallel Text Processing*, pages 1–17. Springer.
- Éric Gaussier, Jean-Michel Renders, Irina Matveeva, Cyril Goutte, and Hervé Déjean. 2004. A geometric view on bilingual lexicon extraction from comparable corpora. In *ACL*, pages 526–533.
- Z.S. Harris. 1954. Distributional structure. *Word*.
- Amir Hazem and Emmanuel Morin. 2012. Adaptive dictionary for bilingual lexicon extraction from comparable corpora. In *Proceedings, 8th international conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, May.
- Myungwon Hwang, Chang Choi, and Pankoo Kim. 2011. Automatic enrichment of semantic relation network and its application to word sense disambiguation. *IEEE Transactions on Knowledge and Data Engineering*, 23:845–858.
- Audrey Larocche and Philippe Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *23rd International Conference on Computational Linguistics (Coling 2010)*, pages 617–625, Beijing, China, Aug.
- Bo Li and Éric Gaussier. 2010. Improving corpus comparability for bilingual lexicon extraction from comparable corpora. In *23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China, Aug.
- Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98*, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Emmanuel Prochasson, Emmanuel Morin, and Kyo Kageura. 2009. Anchor points for bilingual lexicon extraction from small comparable corpora. In *Proceedings, 12th Conference on Machine Translation Summit (MT Summit XII)*, page 284–291, Ottawa, Ontario, Canada.
- Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics, ACL '95*, pages 320–322. Association for Computational Linguistics.
- Raphaël Rubino and Georges Linarès. 2011. A multi-view approach for term translation spotting. In *Computational Linguistics and Intelligent Text Processing*, *Lecture Notes in Computer Science*, pages 29–40.
- Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics, ACL '94*, pages 133–138. Association for Computational Linguistics.

# Predicate Argument Structure Analysis using Partially Annotated Corpora

Koichiro Yoshino, Shinsuke Mori, Tatsuya Kawahara

School of Informatics, Kyoto University  
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan  
yoshino@ar.media.kyoto-u.ac.jp

## Abstract

We present a novel scheme of predicate argument structure analysis that can be trained from partially annotated corpora. In order to allow partial annotation, this semantic role labeler does not require word dependency information. The advantage of partial annotation is that it allows for smooth domain adaptation of training data and improves the adaptability to a variety of domains.

## 1 Introduction

The predicate-argument (P-A) structure is one of the most fundamental and important representations in linguistics (Fillmore, 1968). Many applications use P-A structure as a component, for example, QA systems (Shen and Lapata, 2007), text mining systems (Wang and Zhang, 2009), and a spoken dialogue systems (Yoshino et al., 2011).

P-A structure analysis is regarded as a task of semantic role labeling (SRL). A semantic role represents a meaning of the components in P-A structure (i.e. Propbank (Palmer et al., 2005), FrameNet (Baker et al., 1998), and NAIST Text Corpus (NTC) (Iida et al., 2007b)). Traditional P-A structure analyzers estimate the semantic role labels for an input sentence by referring to a model trained on data annotated with not only semantic role labels but also dependency labels (Surdeanu et al., 2008; Hajič et al., 2009). Most of the previous approaches to P-A structure analysis assume full annotation for P-A structures and the lower layer labels: word boundaries, parts of speech (POS), and dependencies. Given a corpus fully annotated with them, the structural prediction approach was shown to be effective (Watanabe et al., 2010). However, this pre-annotation incurs high annotation costs which prevent us from adapting the analyzer to new domains. Having training data that are representative of a domain is essential for constructing a robust semantic role labeler (Pradhan et al., 2008) because the important information structures are specific to each domain (R.Grishman, 2003). Fully annotated corpus

in target domain is not available in realistic cases, and it is difficult to apply the current supervised approaches to a new domain.

When annotating only the domain-specific area, the use of a partially annotated corpus (Tsuboi et al., 2008; Sassano and Kurohashi, 2010) that allows incomplete annotations improves accuracy efficiently and reduce the number of annotations. The pointwise approach (Neubig and Mori, 2010) enables efficient use of such incomplete language resources in word segmentation tasks and requires only partial annotations for the relevant tasks and lower layer annotations on which they depend. We design a new P-A structure analysis method that enables us to directly estimate semantic role labels by referring to a model that is trained from a corpus that includes only partially annotated tag information without word dependencies.

## 2 Predicate argument structure analysis

In this section, we give a brief explanation of P-A structure and its problems. Then, we describe the typical method of structural prediction for this task based on supervised machine learning.

### 2.1 Predicate-argument (P-A) structure

A predicate-argument (P-A) structure is a relationship between a verbal expression and its arguments, such as the subject, the direct object, and the indirect object. Predicate  $P$  in a document  $D$  has arguments  $A_1, A_2, \dots, A_n$  that have a semantic role  $S_1, S_2, \dots, S_n$ . The notion we used is defined in NAIST Text Corpus (NTC) (Iida et al., 2007b), Japanese text corpora annotated with coreference and P-A relations, which include annotations of subject, direct object, and indirect object. Every case has a property of **depend** or **zero**, and these P-A structure relations are annotated not only predicates, but also event nouns (Komachi et al., 2007).

**Figure 5** shows an example of P-A structures in the NTC. These tags have two properties, one is **depend** or **zero**, the other is *intra* or *inter*. **depend** or **zero** indicates whether or not the argument has a dependency on the predicate, and *intra*

or *inter* indicates whether or not the argument and the predicate exist in the same sentence. NTC also includes annotations of coreference, which we converted into P-A structure tags. As shown in **Figure 1**, P-A structure is located in a higher layer of linguistics that approaches natural language understanding (NLU), and this structure depends on some more basic but much more frequent linguistic phenomena: word boundaries, part of speech (POS), and word dependencies.

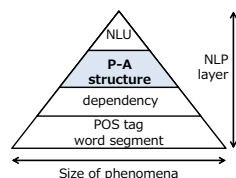


Figure 1: Size of linguistic phenomena.

## 2.2 Typical solution

The typical solution divides the P-A structure prediction into two problems: semantic role labeling (SRL) (Johansson and Nugues, 2008; Björkelund et al., 2009) and zero-anaphora resolution (Iida et al., 2007a; Sasano and Kurohashi, 2009).

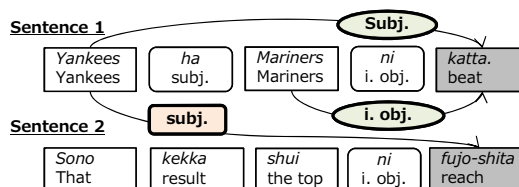
The typical approach requires three preprocessing steps: word segmentation, POS tagging, and dependency parsing. After the preprocessing, the task of SRL improves assigning semantic role labels to the edges in word dependencies. A semantic role labeler performs two tasks: predicate sense estimation, and SRL. Zero-anaphora resolution is treated as an independent problem from the SRL task in the previous research. The zero-anaphora problem is caused by the ellipsis of shared words, and it is a gap in a sentence that has an anaphoric function (Iida et al., 2007a). Some semantic relationships exist in which there is no dependency relationship between their arguments and predicates; this is called zero anaphora.

The task of P-A structure analysis goes beyond the syntactic problem and comes down to a semantic problem to fill in the words that are semantically omitted. Various special approaches can be applied after SRL to solve this problem (Sasano and Kurohashi, 2009; Iida and Poesio, 2011; Hayashibe et al., 2011). Some approaches adopt a *Salience Reference List* (Nariyama, 2002) based on the 1-best argument decision model.

## 2.3 Open problems in P-A structure analysis

Existing approaches require full annotation of word boundaries, POS tags, and word dependencies to use them as features. Most previous approaches to P-A structure tasks assume full annotation for these lower layers. However, the number of linguistic phenomena decreases as we go higher up the NLP layers as shown in Figure 1. Thus, to prepare only one training example for an existing

Translation 1: The Yankees beat the Mariners.



Translation 2: As a result, the Yankees (omitted) reached to the top.

Figure 2: Example of training data made from partially annotated corpus.

P-A structure analyzer, we need to annotate the entire document. To make it worse, these kinds of annotations are costly and difficult for untrained annotators. This difficulty interferes with efficient language resource preparation and reduces domain portability. However, the accuracy of P-A structure analysis increases in accordance with the data size. This indicates that we can realize an improvement just by easily preparing more training data for the target domain document.

## 3 Partial annotation for P-A structures

Partial annotation allows annotators to focus on efficient examples in the target domain document, and to maximize the cost-effectiveness of annotation. For automatic word segmentation and POS tagging, the scheme allows partial annotation of corpus (Tsuboi et al., 2008; Neubig and Mori, 2010; Neubig et al., 2011) and achieves high accuracy and domain portability though annotation of domain-specific areas. Neubig et al. (2011) report that a comparable accuracy to a CRF-based sequential labeling method can be achieved without referring to the estimated labels for unlabeled words. They call this method a pointwise approach. Even with the pointwise assumption, we can estimate labels as accurately as sequential labeling just by referring to the appropriate features.

We design a P-A structure analyzer that directly estimates the semantic role labels by referring to a model that is trained from a corpus. It includes only partially annotated POS tags but not with dependency information for the following reasons. Automatic estimation of POS tags achieves high accuracy in domain adaptation cases, and the annotation cost is small (Neubig et al., 2011), but the accuracy for dependency parsing (Flannery et al., 2011; Sassano and Kurohashi, 2010) is not sufficiently high. However, handcraft annotation cost of dependency is so high, and it disturbs rapid preparation of annotation data.

We show an example of a partially annotated corpus in **Figure 2**. The annotation of “reach” is incomplete, and the information that can be referred to by an analyzer is the fully annotated word boundaries, POS tags, and partially annotated P-A tags. Word boundaries and POS tags are output by

Table 1: Features of SRL:  $w_p$  is a predicate,  $w_a$  is an argument candidate,  $t_i$  is the POS tag of  $w_i$ .

type	feature
word 1-gram	$w_{p-3}, w_{p-2}, w_{p-1}, w_p, w_{p+1}, w_{p+2}, w_{p+3}, w_{a-3}, w_{a-2}, w_{a-1}, w_a, w_{a+1}, w_{a+2}, w_{a+3}$
word 2-gram	$w_{p-1}w_p, w_pw_{p+1}, w_{a-1}w_a, w_a w_{a+1}$
word 3-gram	$w_{p-1}w_pw_{p+1}, w_{a-1}w_a w_{a+1}$
POS 1-gram	$t_{p-3}, t_{p-2}, t_{p-1}, t_p, t_{p+1}, t_{p+2}, t_{p+3}, t_{a-3}, t_{a-2}, t_{a-1}, t_a, t_{a+1}, t_{a+2}, t_{a+3}$
POS 2-gram	$t_{p-1}t_p, t_p t_{p+1}, t_{a-1}t_a, t_a t_{a+1}$
POS 3-gram	$t_{p-1}t_p t_{p+1}, t_{a-1}t_a t_{a+1}$
pairwise	Pairs of POS tags located -2 – +2. Pairs of <b>arg</b> candidate and <b>pred</b> .
distance	Number of <b>pred</b> between the candidate and <b>preds</b>
binary	(1) Closest candidate that has target particle on the right side or not. (2) First candidate that has target particle on the right side or not. (3) The predicate has a slot of target semantic case or not.

a domain-adapted morphological analyzer, and the annotator tags three P-A tags.

## 4 Pointwise P-A structure analysis

In our proposed scheme, syntactic ambiguity resolution and predicate sense disambiguation are not used, in order to achieve easy adaptation. We propose two sequential processes for P-A structure analysis that is trained from partially annotated corpora. Following the discussion in Section 3, we do not assume the dependency structures.

### 4.1 Case existence detection

The first step in the proposed sequential analysis is case existence estimation. The given semantic cases differ according to the type of the predicate. This predicate and semantic case behavior strongly affects the SRL task.

The oracle of the case existence is used for SRL features. For example, the predicate “bet” in Figure 5 contains information indicating that the predicate has two kinds of argument: “subject, zero” and “direct object depend.” We assume that the case existence for each predicate can be estimated with case frames (Kawahara and Kurohashi, 2006). A case frame is a set of a predicate and its potential arguments. It is known that the case frames contribute to the P-A structure analysis performance (Sasano et al., 2008).

### 4.2 SRL and zero-anaphora resolution

The second step is SRL that includes zero-anaphora resolution. We handle the problem with a direct approach for SRL that is redefined as a binary classification problem for the pair of an argument candidate and a predicate. Labeled pairs of argument (**arg**) and predicate (**pred**) are used as positive training example and unlabeled pairs are used as negative training example. In the example of Figure 5, the pair of “fate” and “party” is

a positive example Y, and pairs of “fate” and other candidates are negative examples N.

The features used for classification are listed in **Table 1**. We use simple  $n$ -gram features based on words and POS tags. The pairwise features are POS pairs located at positions from -2 to +2, and pairs of the predicate and the argument candidates. The distance between the argument candidate and the predicate is used as a feature. We used the number of predicates between the predicate and the argument candidate as this feature. Binary features (1) and (2) are based on a previous study on “Centering” theory (Grosz et al., 1995). In this theory, subjects are frequently omitted, and the first candidate tends to be a subject. By contrast, objects are not omitted, and the last candidate tends to be an object. To apply the theory to a pointwise approach, we define features that are independent of syntactic structure. Finally, the result of the processing described above is used as a binary feature (3).

### 4.3 Issues in partial annotation

Two problems arise in applying the classifier to a partially annotated corpus without dependency. First, in existing studies P-A structure analysis leverages a property of the P-A tags **depend** and **zero** (Iida et al., 2007b). Here, **depend** represents that the P-A tag is added on the edge of dependency, and **zero** means the pair of the predicate and argument does not have a relationship of dependency (=zero anaphora). However, it is impossible to use dependency information in our framework, and the attribution makes it difficult to detect the property of the P-A tag. To cope with this problem, we use sentence boundaries, which are trivial in unlabeled documents, for grouping the training set. The other problem is how to create training examples from incomplete annotations. To allow the incomplete annotation perfectly, we incorporate positive examples that are clearly annotated.

## 5 Evaluations

We conducted three experiments to evaluate the proposed method: SRL, corpus size discrimination, and domain adaptation.

### 5.1 Experimental settings

We use the NTC (Iida et al., 2007b) which is annotated with P-A relations and coreferences. The NTC is constructed from Japanese newspaper articles, and has two domains: news and editorials. In the NTC, there are three different types of annotation on pairs of predicates and their arguments: subject, direct object, and indirect object. Every tag has a property of **depend** or **zero**. The

Table 2: Results of P-A analysis (with case frame, using the property of depend and zero ).

	role label	prec.	recall	F
dep.	subject	0.747	0.754	0.750
	d. obj.	0.908	0.930	0.919
	i. obj.	0.953	0.947	0.950
	total	0.839	0.849	0.844
	total (w.o. feat. (3))	0.744	0.683	0.712
zero	subject	0.305	0.120	0.172
	d. obj.	0.560	0.212	0.307
	i. obj.	0.402	0.127	0.192
	total	0.402	0.127	0.192
	total (w.o. feat. (3))	0.251	0.115	0.157
total		0.580	0.321	0.413
cf. (zero)	subject	0.265	0.302	0.282
	d. obj.	0.092	0.129	0.107
	i. obj.	0.048	0.041	0.044

Table 3: Results of P-A analysis (with case frame, using the property of *intra* and *inter* ).

	role label	prec.	recall	F
<i>intra</i>	subject	0.624	0.520	0.567
	d. obj.	0.841	0.809	0.825
	i. obj.	0.868	0.807	0.836
	total	0.730	0.646	0.686
<i>inter</i>	subject	0.311	0.118	0.171
	d. obj.	0.320	0.048	0.083
	i. obj.	0.329	0.085	0.135
	total	0.312	0.111	0.164
total		<b>0.602</b>	<b>0.366</b>	<b>0.455</b>
cf. ( <i>inter</i> )	subject	0.221	0.273	0.244
	d. obj.	0.050	0.101	0.066
	i. obj.	0.030	0.023	0.026

NTC has lower layer annotations: word boundaries, POS, and segment-based dependencies. We used the word segments and POS tags as-is, and constructed P-A classifiers. We evaluated the proposed SRL in the newspaper article domain. We used linear support vector machine (SVM) (Fan et al., 2008) with the one-versus-rest method, by using the features described in Table 1.

## 5.2 Evaluation of SRL

The results using 5-fold cross validation are listed in **Table 2** and **3**. Evaluations that are classified with the existing *depend* and *zero* property are given in Table 2. Classifiers used in “w.o. feat. (3)” do not refer to case existence features (the binary feature (3) in Table 1). We can see that case frames play a large role in improving the labeling accuracy. This *depend* and *zero* property is based on the dependency, which cannot be referred to in the pointwise approach. As an alternative, we used sentence boundaries for the tag classification and Table 3 shows the result. The bottom “cf.” rows in Tables 2 and 3 are the result of the previous work (Sasano and Kurohashi, 2011) for comparison<sup>1</sup>. In the comparison, the accuracies of our work are comparable to the accuracies of the previous work. By comparing the total F mea-

<sup>1</sup>Sasano and Kurohashi discussed this task, but the article is written in Japanese. They evaluated the accuracy for zero anaphora in two models: *intra* and *inter*, and we calculated the weighted mean of them for fair comparison.

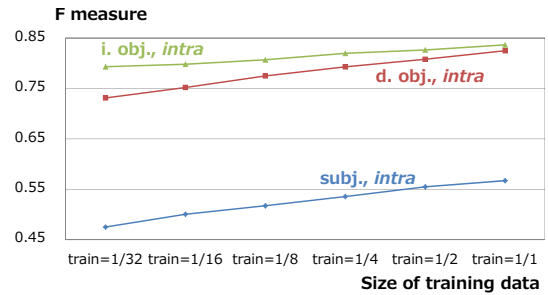


Figure 3: Effect of corpus size in *intra* case.

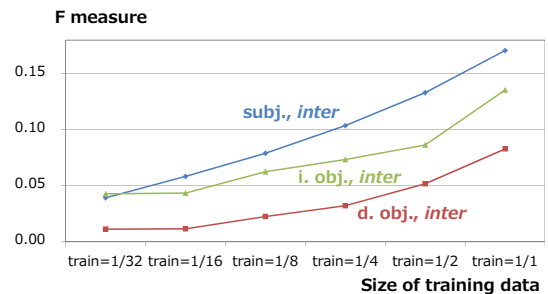


Figure 4: Effect of corpus size in *inter* case.

sure in Table 2 (0.413) and that in Table 3 (0.455), we can say that this *intra* and *inter* property works better than the *depend* and *zero* property in our pointwise classifier.

## 5.3 Effect of corpus size

We show the relationship between the training corpus size and the accuracy in **Figures 3** and **4**. The horizontal axes of these graphs are the log-scaled corpus size. The graphs show that P-A structure analysis accuracy increases linearly in proportion to the log-scaled data size and do not saturate. This result supports our framework of efficient resource usage.

## 6 Conclusion

We presented a novel scheme of P-A structure analysis based on the pointwise approach that makes it possible to use partially annotated corpora. This paper can be seen as an extension of the pointwise approach to a higher NLP layer that allows us to concentrate annotation work on the focused task. The results indicated that our scheme reduces the cost of constructing language resource and makes it easy to adapt the P-A structure analyzer while maintaining comparable accuracy to current analysis frameworks.

In future work, we plan to evaluate our pointwise P-A resolution method in the domain adaptation case in terms of personal costs of annotation and investigate improving accuracy by using other estimated information.



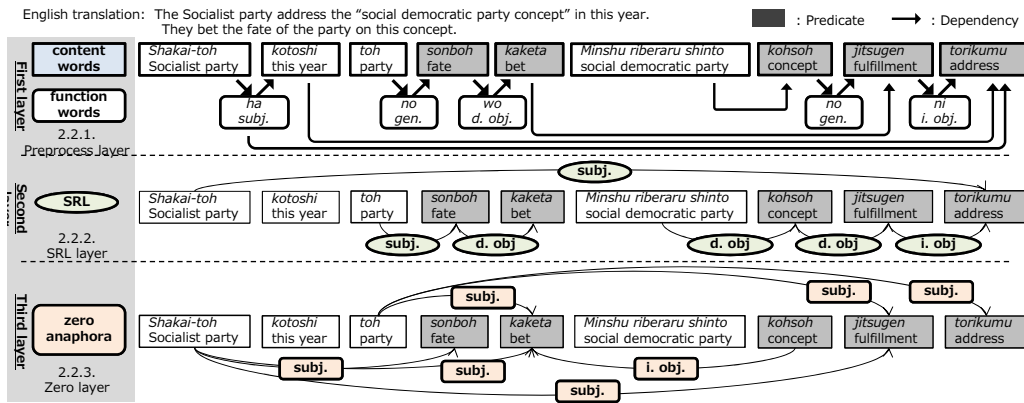


Figure 5: Example of P-A structure analysis.

## A Figure 5 shows an example of P-A

### References

- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proc. ACL-COLING*, pages 86–90.
- Anders Björkelund, Love Hafdel, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proc. CoNLL: Shared Task*, pages 43–48.
- 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(4):1871–1874.
- Charles J. Fillmore. 1968. The case for case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*.
- Daniel Flannery, Yusuke Miyao, Graham Neubig, and Shinsuke Mori. 2011. Training dependency parsers from partially annotated corpora. In *Proc. IJCNLP*, pages 776–784.
- Hagen Fürstenaу and Mirella Lapata. 2009. Semi-supervised semantic role labeling. In *Proc. EACL*, pages 220–228.
- Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.
- Jan Hajič et al. 2009. The conll-2009 shared task: syntactic and semantic dependencies in multiple languages. In *Proc. CoNLL: Shared Task, CoNLL '09*, pages 1–18.
- Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. 2011. Japanese predicate argument structure analysis exploiting argument position and type. In *Proc. IJCNLP*, pages 201–209.
- Ryu Iida and Massimo Poesio. 2011. A cross-lingual ilp solution to zero anaphora resolution. In *Proc. ACL-HLT*, pages 804–813.
- Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2007a. Zero-anaphora resolution by learning rich syntactic pattern features. *ACM Transactions on Asian Language Information Processing (TALIP)*, 6(4):12:1–12:22.
- Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007b. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proc. the Linguistic Annotation Workshop*, pages 132–139.
- Richard Johansson and Pierre Nugues. 2008. Dependency-based semantic role labeling of PropBank. In *Proc. EMNLP*, pages 69–78.
- Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proc. HLT-NACCL*, pages 176–183.
- Daisuke Kawahara, Sadao Kurohashi, and Koiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proc. LREC*, pages 2008–2013.
- Mamoru Komachi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2007. Learning based argument structure analysis of event-nouns in japanese. In *Proc. PACLING*, pages 120–128.
- Shinsuke Mori and Graham Neubig. 2011. A pointwise approach to pronunciation estimation for a tts front-end. In *Proc. INTERSPEECH*, pages 2181–2184.
- Shigeko Nariyama. 2002. Grammar for ellipsis resolution in japanese. In *Proc. TMI*, pages 135–145.
- Graham Neubig and Shinsuke Mori. 2010. Word-based partial annotation for efficient corpus construction. In *Proc. LREC*.
- Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. ACL*, pages 529–533.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Sameer S. Pradhan, Wayne Ward, and James H. Martin. 2008. Towards robust semantic role labeling. *Computational Linguistics*, 34(2):289–310, jun.
- R. Grishman. 2003. Discovery methods for information extraction. In *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, pages 243–247.
- Ryohei Sasano and Sadao Kurohashi. 2009. A probabilistic model for associative anaphora resolution. In *Proc. EMNLP*, pages 1455–1464.
- Ryohei Sasano and Sadao Kurohashi. 2011. A discriminative approach to japanese zero anaphora resolution with large-scale case frame. *Journal of Information Processing (in Japanese)*, 52(12):3328–3337.
- Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2008. A fully-lexicalized probabilistic model for japanese zero anaphora resolution. In *Proc. COLING*, pages 769–776.
- Manabu Sassano and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for japanese dependency parsing. In *Proc. ACL*, pages 356–365.
- Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proc. EMNLP-CoNLL*, pages 12–21.
- Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. CoNLL: Shared Task*, pages 159–177.
- Ivan Titov and Alexandre Klementiev. 2012. Semi-supervised semantic role labeling: Approaching from an unsupervised perspective. In *Proc. COLING*, pages 2635–2652.
- Yuta Tsuboi, Hisashi Kashima, Hiroki Oda, Shinsuke Mori, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proc. COLING*, pages 897–904.
- Rui Wang and Yi Zhang. 2009. Recognizing textual relatedness with predicate-argument structure. In *Proc. EMNLP*, pages 784–792.
- Yofaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 2010. A structured model for joint learning of argument roles and predicate senses. In *Proc. ACL*, pages 98–102.
- Koichiro Yoshino, Shinsuke Mori, and Tatsuya Kawahara. 2011. Spoken dialogue system based on information extraction using similarity of predicate argument structures. In *Proc. SIGDIAL*, pages 59–66.
- Koichiro Yoshino, Shinsuke Mori, and Tatsuya Kawahara. 2012. Language modeling for spoken dialogue system based on filtering using predicate-argument structures. In *Proc. COLING*, pages 2993–3002.

# Statistical Dialogue Management using Intention Dependency Graph

Koichiro Yoshino<sup>1,2</sup>, Shinji Watanabe<sup>1</sup>, Jonathan Le Roux<sup>1</sup>, John R. Hershey<sup>1</sup>

<sup>1</sup>Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA, 02139, USA  
{watanabe, leroux, hershey}@merl.com

<sup>2</sup>School of Informatics, Kyoto University, Sakyo, Kyoto, 606-8501, Japan  
yoshino@ar.media.kyoto-u.ac.jp

## Abstract

We present a method of statistical dialogue management using a directed intention dependency graph (IDG) in a partially observable Markov decision process (POMDP) framework. The transition probabilities in this model involve information derived from a hierarchical graph of intentions. In this way, we combine the deterministic graph structure of a conventional rule-based system with a statistical dialogue framework. The IDG also provides a reasonable constraint on a user simulation model, which is used when learning a policy function in POMDP and dialogue evaluation. Thus, this method converts a conventional dialogue manager to a statistical dialogue manager that utilizes task domain knowledge without annotated dialogue data.

## 1 Introduction

Statistical approaches based on reinforcement learning, such as the Markov decision process (MDP) and partially observable Markov decision process (POMDP), have been successfully applied to dialogue management (Levin et al., 2000; Williams and Young, 2007; Li, 2012). These approaches allow us to consider all possible future actions of a dialogue system, and thus to obtain a new optimal dialogue strategy which could not be anticipated in conventional hand-crafted dialogue systems. Moreover, the statistical dialogue framework can be combined with conventional rule-based dialogue management in hybrid systems, (Williams, 2008; Lee et al., 2010), which combine the optimal dialogue strategy in the statistical approach with the lower cost of data and maintenance of the rule-based approach.

Our research focuses on a practical application of a hybrid statistical dialogue management based

on POMDP to conventional rule-based dialogue management via the use of an intention dependency graph (IDG). The IDG derives from the conventional rule-based dialogue system (Dahl et al., 1994; Bohus and Rudnicky, 2003), and it constrains the transition matrix and provides a user simulation as a substitute for dialogue data.

The object of POMDP optimization is to produce a policy that maps from user states to system actions such that the overall expected cost of the dialogue is minimized. Such optimization typically requires data from dialogue corpora, which are manually annotated with task-oriented dialogue-act tags. On the other hand, the benefit of a hybrid approach is that human domain knowledge can be used to constrain the possible user states in the dialogue manager. We follow this idea by using an IDG, which expresses the task-domain knowledge through a directed graph of states from more general intention categories to more specific parameters of the intention categories. **Figure 1** shows an example of such a graph, where each node is associated with a (potentially partial) user intention. In previous studies, this kind of domain knowledge is used to restrict the user state and system action state space (Lemon et al., 2006; Williams, 2008; Young et al., 2010; Varges et al., 2011). However, our approach does not restrict the possible system action states, but transfers the information structure to the definition of user simulation and state transition probabilities. The system is allowed to consider all possible system actions by following the user states that reflect the IDG.

## 2 Statistical dialogue management

The main random variables involved at a dialogue turn  $t$  are as follows.  $s^t = i \in \mathcal{I}_s$  is the hidden true user state at turn  $t$ . It is constrained by the hidden user goal  $g \in \mathcal{I}_g$  and the true user state at the previous turn.  $o^t = l \in \mathcal{I}_s$  is the observation

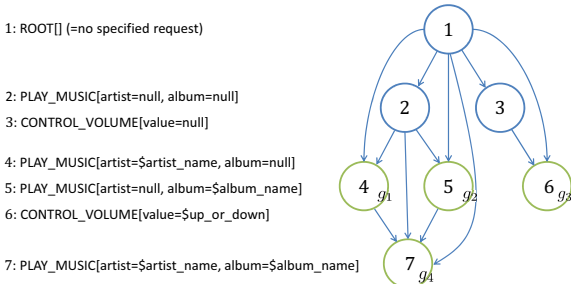


Figure 1: An example of a directed intention dependency graph.

of the user state by the system. It includes errors caused by automatic speech recognition (ASR), natural language understanding (NLU) and intention understanding (IU). Uncertainty on the observation  $o^t$  caused by errors in the preprocessor (ASR, NLU, and IU) is encompassed in the conditional probability  $O_{li}^t = p(o^t = l | s^t = i)$ .  $a^t = k \in \mathcal{K}$  is the system action.  $\hat{k}$  is the optimal system action that is acquired in the learning step. The goal of statistical dialogue management is to output an optimal system action  $\hat{a}^t = \hat{k}$  given an observation  $o^t$ , based on the probability of  $s^t$  in a soft decision manner. The probability of the user state  $s^t$  given an observation sequence  $o^{1:t}$  from 1 to  $t$  with confidence  $O^{1:t}$  is denoted by  $b_i^t = p(s^t = i | o^{1:t}; O^{1:t})$ , and referred to as ‘‘belief’’. To avoid clutter, we will usually omit  $O^{1:t}$ .

## 2.1 Belief update

We consider a belief update equation based on the graphical model shown in **Figure 2**, assuming that the system actions  $a^{1:t}$  are given. We can obtain the following update equation from  $b_i^t$  to  $b_{i'}^{t+1}$ :

$$b_{i'}^{t+1} = p(s^{t+1} = i' | o^{1:t+1}) \quad (1)$$

$$\propto \sum_i p(o^{t+1}, i' | i) (b_i^t)^\beta, \quad (2)$$

where  $\beta$  is a forgetting factor for the belief, and  $0 \leq \beta \leq 1$ . Then, by introducing the system action  $a^t = k$  based on the sum rule, we can rewrite  $p(o^{t+1}, i' | i)$  in Eq. (2) as follows:

$$\begin{aligned} \sum_k p(o^{t+1}, i', k | i) &= \sum_k p(o^{t+1}, i' | i, k) \delta_{\hat{k}k} \\ &= p(o^{t+1} | i') p(i' | i, \hat{k}) \end{aligned} \quad (3)$$

where  $p(k | i) = \delta_{\hat{k}k}$  is obtained by the decision making step in the POMDP. We rewrite the distributions in Eq. (3) as follows:  $p(i' | i, \hat{k}) = T_{ii'\hat{k}}$

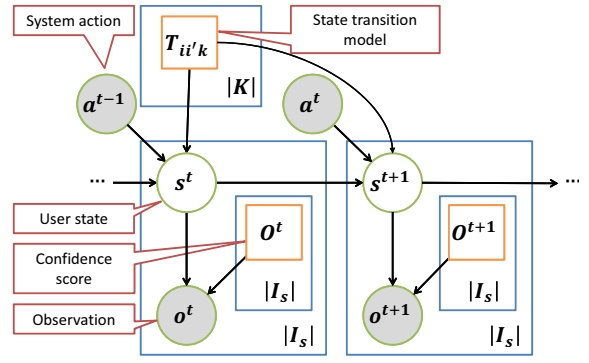


Figure 2: Graphical model of user state sequences given system actions  $a^{t-1}$  and  $a^t$ . This graphical model shows user behavior that is observed from the system.

and  $p(o^{t+1} = l | i') = O_{li'}^{t+1}$ .  $T_{ii'\hat{k}}$  are the user state transition probabilities given system action  $\hat{k}$ , and  $O_{li'}^{t+1}$  are the confidence scores given by the pre-processor. In conventional studies, the state transition probabilities  $T_{ii'\hat{k}}$  are learned from annotated data. In our scheme, the probabilities can be obtained by using the IDG, as described in Section 3.4. We finally obtain

$$b_{i'}^{t+1} \propto O_{li'}^{t+1} \sum_i T_{ii'\hat{k}} (b_i^t)^\beta. \quad (4)$$

Once the system estimates the belief  $b_i^t$ , it can output the optimal action  $\hat{a}^t$  as  $\hat{a}^t = \pi^*(\{b_i^t\}_{i=1}^{|I_s|})$ .  $\pi$  is called a policy function, and  $\pi^*$  is an optimal policy function pre-computed in the learning step described in the following Section.

## 2.2 Learning step

The aim of the learning step in reinforcement learning is to acquire the best policy  $\pi^*$ . Many algorithms formulated to solve the reinforcement learning problem have been proposed (Shani et al., 2013). While most advanced algorithms require transition probabilities  $T_{ii'\hat{k}}$  that are calculated using annotated corpora, our approach aims at learning a POMDP without any data. We thus use one of the most basic algorithms, Q-learning (Watkins and Dayan, 1992), as it can acquire the policy without using transition probabilities. Q-learning relies on the estimation of a Q-function  $Q(b^t, a^t)$ , which computes the expected future reward of a system action  $a^t$  at dialogue turn  $t$  given the current belief  $b^t = \{b_i^t\}_{i=1}^{|I_s|}$  of the user state. The Q-function can be obtained by iterative updates on training dialogue data. The up-



dates do not involve the transition probabilities  $T_{ii'k}$ , thus we can acquire the optimal policy without requiring knowledge of this function. Given the Q-function, the optimal policy is determined as  $\pi^*(b^t) = \arg \max_{a^t} Q(b^t, a^t)$ .

### 3 Dialogue management using intention dependency graph

#### 3.1 Intention dependency graph

An intention dependency graph (IDG) is a representation of a user’s intention in a hierarchy, with broad categories of the intention at the top, and specific instantiations of those categories at the bottom, as shown in **Figure 1**. A child node in the graph represents a more specific intention than the parent node, so that the flow from top to bottom represents the completion of the full specification of an intention. However, the graph is not necessarily a tree, and hence there may be multiple paths from a parent node to any descendent node. A node that is fully specified and actionable by the system can be considered a user goal. In node 7, which is a child of node 2, both the album and artist are specified, and the system has enough information to perform the desired action. Such a graph is automatically generated from task knowledge that is usually designed by hand for a conventional rule-based dialogue manager (Dahl et al., 1994; Bohus and Rudnicky, 2003), as a graphical user interface, and can be obtained by forming a taxonomy of the possible system actions. In our context, a node in this graph represents a hypothesis of the user’s intention and/or goal.

#### 3.2 User simulator

Training a statistical dialogue management system in the absence of large amounts of dialogue data requires a user simulator to ensure adequate coverage of possible user states. In a general dialogue, the system action and the user state would follow a dialogue history and lead toward a user goal. The simulator thus samples user states  $s^{t+1} = i$ , at every time step, tending toward a user goal  $g$ , and depending on the previous system action  $a^t = k$ . Thus, our approach defines the sampling distribution  $p(i|g, k)$  by using IDG. Our approach gives uniform distribution to hypotheses that are outputted by the IDG. We show an example IDG in **Figure 1** and a dialogue example in **Figure 6** of Appendix.

#### 3.3 Learning without any annotated data

We discuss the learning for the POMDP that uses our IDG. In our task, no data can be referred to and we cannot calculate the transition probability that is generally calculated from an annotated data for the belief update. This property makes it impossible to establish the exact value of the state-value function. In standard POMDP learning, sampling belief point approaches that select a small set of representative belief points such as point-based value iteration (PBVI) can be applied (Pineau et al., 2003). However, it is difficult to sample a small set of belief points without any tagged data. Therefore, we calculate the action-value function  $Q(b^t, a^t)$ , and simulate the noise with a grid-based approach (Lovejoy, 1991; Bonet, 2002). The grid-based approach can select points in accordance with a grid and a noise parameter  $\eta$  that is released from the data. In our learning approach, a sample of belief  $b_i^t = p(s^t = i|o^{1:t}, O^{1:t})$  is given by  $p(o^{t+1} = l|i') = O_{li'}^{t+1}$  where

$$O_{li'}^{t+1} = \begin{cases} 1 - \eta & l = i' \\ \frac{\eta}{|\mathcal{I}_s|-1} & l \neq i'. \end{cases} \quad (5)$$

We tried noise  $\eta = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ . The resulting policy does not reflect the belief update, but we can use the belief update method that follows the IDG.

#### 3.4 State transition and belief update

The state transition probability  $T_{ii'k}$  is one of the most important components of the belief update in the POMDP framework. To obtain the transition probabilities, we usually require user state and system action data with annotated tags. However, we cannot calculate the probability because of the lack of annotated data. Therefore, we define the state transition probability by using an IDG similar to user simulation, as discussed in Section 3.2. By employing time-invariant user goal  $g$ , time-variant user state  $s^t = i$  and time-variant best system action  $a^t = \hat{k}$  in Section 2, we can represent the state transition probabilities, as follows:

$$p(i'|i, \hat{k}) = \sum_g p(i'|g, i, \hat{k})p(g|i, \hat{k}) \quad (6)$$

We approximate  $p(i'|g, i, \hat{k})$  by user simulator  $p(i'|g, \hat{k})$ . This means that the next user state  $i'$  does not depend on the previous user state  $i$ . We approximate  $p(g|i, \hat{k}) \simeq p(g|i)$  because the user

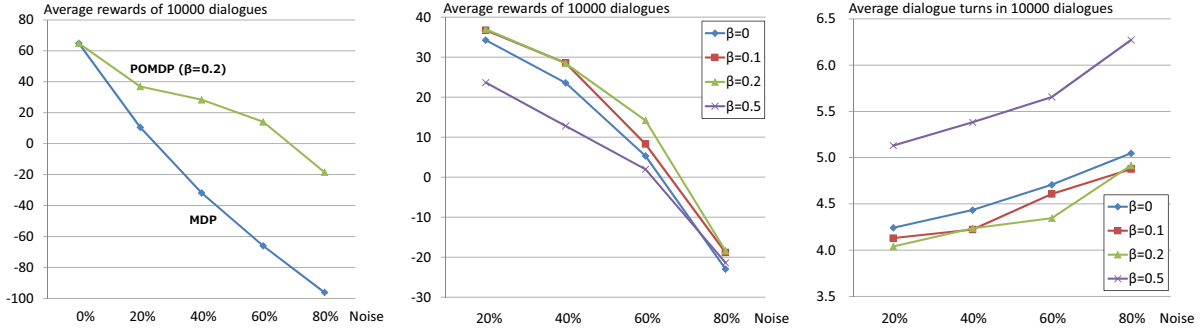


Figure 3: Average rewards of Figure 4: The effect of forgetting Figure 5: The effect of forgetting 10000 dialogues between the ob- factor  $\beta$  as regards the average re- factor  $\beta$  as regards the average di- tained dialogue manager and the wards of 10000 dialogues. al- ogue turns of 10000 dialogues. user simulator.

goal  $g$  can be estimated from the user state  $i$  by using the IDG. As a result, Eq. (6) is approximated as,

$$(6) \cong \sum_g \underbrace{p(i'|g, \hat{k})}_{\text{simulator}} \underbrace{p(g|i)}_{\text{goal model}} \quad (7)$$

Here, **simulator** is the user simulator that is defined in Section 3.2, and **goal model** is a goal estimation model that can be calculated from an IDG. Our user simulator does not perform in accordance with  $p(s^{t+1} = i'|g, a^t = \hat{k})$  exactly, but our model uses the track back of the user simulator that is defined in Section 3.2. The probability of a goal estimation model is defined as  $p(g|i)$ , which expresses possible goals given a user state  $s^t = i$ .

## 4 Evaluations

We evaluate our statistical dialogue management approach, which uses the IDG. These are experimental evaluations with the user simulator that follows Section 3.2. In the experiment, we used an IDG that had 957 states including 667 goals.

### 4.1 Evaluation of average reward

We evaluated dialogue managers in terms of the average reward for 10000 dialogues between the user simulator and the obtained dialogue manager. We simulated uniformly distributed noises that are defined on Eq. (5) for observation. We tried six grids that suppose a uniform distribution given by Eq. (5). The parameters ( $\eta = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ ) were sampled in the Q-learning of the POMDP. We used parameters  $\gamma = 0.8$  and  $\epsilon = 0.2$ . The belief update defined in Section 3.4 was used for the dialogue evaluation. For comparison, we prepared an MDP based

dialogue manager that learned from observations without any noise. The average rewards result is shown in **Figure 3**. In this experimental result, the POMDP dialogue manager performed better than the MDP based dialogue manager (MDP) in noisy cases. The effects of forgetting factor  $\beta$  in terms of average reward and average dialogue turn are shown in **Figure 4** and **Figure 5**. In this graph, the proposed POMDP framework, which includes state transition probabilities, works best at the point  $\beta = 0.2$ . These figures show that the approach depended on the forgetting factor and the robust setting of  $\beta$  is left to future work.

**Figure 7** in Appendix shows an example of dialogue between the user simulator and the dialogue manager. This example was obtained with  $\eta = 0.8, \beta = 0.2$ .

## 5 Conclusion and discussion

We have proposed a dialogue management framework that uses a directed IDG. The IDG is hand-crafted during the construction of the conventional rule-based dialogue system, and our approach can easily adapt rule-based systems to a statistical dialogue management framework. The proposed framework does not require annotated dialogue data in the initial deployment that are essential for the typical statistical dialogue management framework, and this enables rapid and easy adaptation. The proposed scheme is developed purely based on a probability process, and the framework can be extended to use annotated data to estimate model parameters, which will be future work. Ongoing work includes evaluation with real user or realistic user simulator that is constructed from dialogue logs.

## References

- Dan Bohus and Alexander I. Rudnicky. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proc. of EUROSPEECH*.
- Blai Bonet. 2002. An e-optimal grid-based algorithm for partially observable Markov decision processes. In *Proc. of ICML*, pages 51–58.
- Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Proc. of the workshop on Human Language Technology*, pages 43–48.
- Lucie Daubigny, Matthieu Geist, and Olivier Pietquin. 2012. Off-policy learning in large-scale POMDP-based dialogue systems. In *IEEE-ICASSP*, pages 4989–4992.
- M. Gašić, F. Jurčićek, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young. 2010. Gaussian processes for fast policy optimisation of POMDP-based dialogue managers. In *Proc. of SIGDIAL*, pages 201–204.
- F. Jurcicek, B. Thomson, S. Keizer, F. Mairesse, M. Gasic, K. Yu, and S. Young. 2010. Natural belief-critic: a reinforcement algorithm for parameter estimation in statistical spoken dialogue systems. In *Proc. of INTERSPEECH*.
- Cheongjae Lee, Sangkeun Jung, Kyungduk Kim, Donghyeon Lee, and Gary Geunbae Lee. 2010. Recent approaches to dialogue management for spoken dialog systems. *Journal of Computer Science and Engineering*, 4(1):1–22.
- Oliver Lemon, Xingkun Liu, Daniel Shapiro, and Carl Tollerander. 2006. Hierarchical reinforcement learning of dialogue policies in a development environment for dialogue systems: Reall-dude. In *Proc. of the 10th Workshop on the Semantics and Pragmatics of Dialogue*, pages 185–186.
- Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *Speech and Audio Processing, IEEE Transactions on*, 8(1):11–23.
- William Li. 2012. Understanding user state and preferences for robust spoken dialog systems and location-aware assistive technology. Master’s thesis, Massachusetts Institute of Technology.
- William S Lovejoy. 1991. Computationally feasible bounds for partially observed Markov decision processes. *Operations research*, 39(1):162–175.
- Teruhisa Misu and Hideki Kashioka. 2012. Simultaneous feature selection and parameter optimization for training of dialog policy by reinforcement learning. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 1–6. IEEE.
- George E Monahan. 1982. State of the art? a survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1–16.
- Sébastien Paquet, Ludovic Tobin, and Brahim Chaib-draa. 2005. An online POMDP algorithm for complex multi-agent environments. In *Proc. of AAMAS*, pages 970–977.
- Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. of IJCAI*, volume 18, pages 1025–1032. LAWRENCE ERLBAUM ASSOCIATES LTD.
- ShaoWei Png and Joelle Pineau. 2011. Bayesian reinforcement learning for POMDP-based dialogue systems. In *Proc. of IEEE-ICASSP*, pages 2156–2159. IEEE.
- Guy Shani, Joelle Pineau, and Robert Kaplow. 2013. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51.

- Matthijs TJ Spaan and Nikos Vlassis. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of artificial intelligence research*, 24(1):195–220.
- Sebastian Varges, Giuseppe Riccardi, Silvia Quarteroni, and Alexei V Ivanov. 2011. POMDP concept policies and task structures for hybrid dialog management. In *Proc. of IEEE-ICASSP*, pages 5592–5595.
- Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8(3):279–292.
- Jason D Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.
- Jason D. Williams. 2008. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *Proc. of INTERSPEECH*.
- 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 & Language*, 24(2):150–174.

## A Dialogue examples

```

g = Goal: 7 = PLAY_MUSIC(artist=The Beatles, album= Abbey Road)

s0 = 1 : ROOT[]
Ask question on possible goals from 1: {Do: 1 = "What do you want me to do?"}
a0 = Do: 1 = "What do you want me to do?"
-----
s1 = 4 : "Play The Beatles" (mumbled)
ASR/NLU/IU output: "Play $unknown_slot" ← ASR mistake
o1 = 2 : PLAY_MUSIC(artist=NULL, album=NULL)
Launch a possible command from 2: {Do: 2 = "Please say album and/or artist."
Confirm: 2 = "Do you want to play music?"}
a1 = Do: 2 = "Please say album and/or artist" ← System selected Do: 2
s2 = 4 : "Play The Beatles" (clearer)
ASR output: "Play The Beatles"; NLU/IU output: "Play $artist=[The Beatles]"
o2 = 4 : PLAY_MUSIC(artist=The Beatles, album=NULL)
Launch a possible command from 4: {Do: 4 = "Please say specific album."
Goal: 4 = "I will play all albums of The Beatles."
Confirm: 4 = "Do you want to play The Beatles?"}
a2 = Do: 4 = "Please say specific album" ← System selected Do: 4
-----
s3 = 7 : "Play Abbey Road by The Beatles"
ASR/NLU/IU output: "Play $album=[Abbey Road] by $artist=[The Beatles]"
o3 = 7 : PLAY_MUSIC(artist=The Beatles, album= Abbey Road)
Launch a possible command from 7: {Goal: 7 = "I will play Abbey Road by The Beatles."
Confirm: 7 = "Do you want to play Abbey Road by The Beatles?"}
a3 = Goal: 7 = "I will play Abbey Road by The Beatles" ← System selected Goal: 7

```

Figure 6: A dialogue example.

User Simulator	System (Dialogue Manager)
Draw $g = \text{Goal: } i_6$ Ask $s^0 = i$ with $P(i g)$ $s^0 = i_3$ selected	Recognize $s^0 = i_3$ in conf. 0.2 Respond $a^0 = \text{Confirm: } i_3$ , Belief point [ $s^0 = i_3$ conf.=0.2]
Ask $s^1 = i$ with $P(i g, k)$ $s^0 = i_3$ selected	Recognize $s^1 = i_3$ in conf. 0.2 Update belief $s^1 = i_3$ in conf. 0.8229 Respond $a^1 = \text{Do: } i_3$ , Belief point [ $s^1 = i_3$ conf.=0.8]
Ask $s^2 = i$ with $P(i g, k)$ $s^0 = i_6$ selected	Recognize $s^2 = i_6$ in conf. 0.2 Update belief $s^2 = i_6$ in conf. 0.4303 Respond $a^2 = \text{Goal: } i_6$ , Belief point [ $s^2 = i_6$ conf.=0.4]

Figure 7: An example of the obtained dialogue between the user simulator and the our system.

# Repairing Incorrect Translation with Examples

Junguo Zhu, Muyun Yang, Sheng Li, Tiejun Zhao

School of Computer Science and Technology, Harbin Institute of Technology  
Harbin, China

{ymy, jgzhu}@mtlab.hit.edu.cn; {lisheng, tjzhao}@hit.edu.cn

## Abstract

This paper proposes an example driven approach to improve the quality of MT system outputs. Specifically, We extend the system combination method in SMT to combine the examples by two strategies: 1) estimating the confidence of examples by the similarity between source input and the source part of examples; 2) approximating target word posterior probability by the word alignments of the bilingual examples. Experimental results show a significant improvement of 0.64 BLEU score as compared to one online translation service (Google Translate).

## 1 Introduction

Statistical Machine Translation (SMT), state-of-the-art solution, has remarkable success with the support of the large-scale bilingual corpora to boost the translation quality at present. However, due to the long tail effect of human language, statistical anomalies in the training data can cause that tons of desired translation knowledge could not be statistically learned from the large-scale bilingual corpora. As a result, bulks of the specific translation requirements not well addressed still perplex machine translation academia and industry.

Combining the examples with machine translations output is a good solution to improve translation quality for this issue. Several methods have been proposed in recent years. One approach tries to replace relevant chunks, taking advantage of Translation Memory (TM). Its motivation is to store and to retrieve similar translation examples for a given input, then to avail of examples to replace the similar chunks into the input by the threshold of similar score (Smith and Clark, 2009; Koehn and Senellart, 2010) or by the decision of an automatic classifier (He et al., 2010; Ma et al., 2011). Another approach

tries to enhance phrase table of SMT, integrating collected bilingual pairs into the phrase table (Biçici and Clark, 2009; Simardand Isabelle, 2010; DauméIII and Jagarlamudi, 2011).

Different to the above studies in which the EBMT and SMT function in a pipeline style, the work in this paper tries to integrate the SMT results and translation examples in a unified framework. In parallel to the system combination in SMT, we try to integrate the translation examples into the confusion network, allowing each word in both SMT results and examples to compete for the optimal output. In order to achieve the goal, the proposed method introduces some new features to bridge the statistical and example translation.

This paper presents an approach to repairing the translation errors via retrieving translation examples from examples corpus. The effectiveness of our method is validated on the standard test set of Olympics task in IWSLT 2012 Evaluation Campaign. Experimental results show that an absolute increase of 0.64 BLEU score is observed after repairing original translations. This significant improvement suggests the proposed strategy as a promising solution to the subtle task of integrating example knowledge into statistical model outputs, as well as a practical way to boost current MT service.

## 2 Repairing Translations with Improved Confusion Network

Repairing translation can be viewed as a process of translation knowledge fusion. As illustrated in Fig.1, the proposed approach consists of following steps. We first obtain an online translation system output  $E_0$  for a given input sentence  $F$ . Then we retrieve the top- $n$  examples  $\{ \langle ex\_F_i, ex\_E_i \rangle \mid (i \in \{1, 2, \dots, n\}) \}$  from bilingual examples corpus which are most similar to  $F$ . Then taking the translation  $E_0$  as the initial skele-

ton, we construct a confusion network by adding the top- $n$  examples into the skeleton incrementally by the word alignments relation between the current skeleton and the  $i$ -th example  $ex_{E_i}$ . The key step is to estimating the word confidence. In this work, we design a feature based on example confidence and word posterior probability by word alignment of examples. Finally, we decode the confusion network by the classic features used in MT combination and new features via a log-linear model.

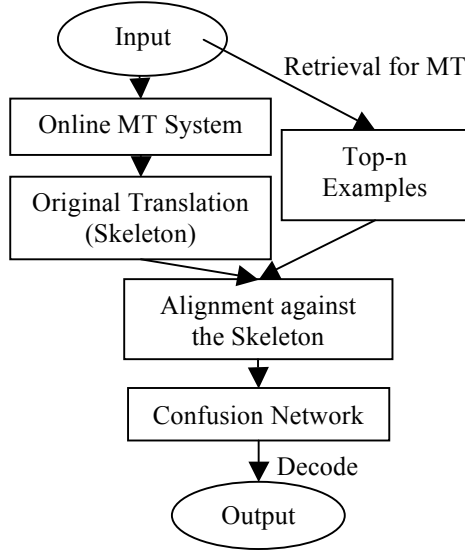


Figure 1. Framework of Repairing Incorrect Translation by Examples

## 2.1 Estimating Example Confidence

We use a word-based vector space model to retrieve examples from bilingual corpus, by comparing the deviation of angles between the source part of each example vector  $ex_{F_i}$  and the source input vector  $F$ . The Cosine Similarity of the vectors is applied to measure the similarity between the source input sentence  $F$  and the each  $plex_{F_i}$ , as calculated by:

$$CosSim(ex_{F_i}, F) = \frac{ex_{F_i} \cdot F}{\|ex_{F_i}\| * \|F\|} \quad (1)$$

where  $ex_{F_i} \cdot F$  is the intersection between the vector  $ex_{F_i}$  and the vector  $F$ .  $\|ex_{F_i}\|$  is the norm of the vector  $ex_{F_i}$  and  $\|F\|$  is the norm of the vector  $F$ . To balance the word recall and precision of examples, we filter the examples by simply keeping top- $n$  similar examples for fusion.

Obviously, a reasonable assumption is that the target example has a higher confidence to occur in the final output if the corresponding source part of example has a higher similarity with the source input sentence. Therefore, we estimate the

confidence  $C_i$  of each target example  $ex_{E_i}$  ( $i \in \{1, 2, \dots, n\}$ ) by the Cosine Similarity score between the source part of example  $ex_{F_i}$  and the source input  $F$ .

## 2.2 Estimating Word Posterior Probability by Word Alignment of Examples

To penalize the irrelevant information from examples, we estimate word posterior probability by word alignment between source words and target words in examples. For word alignment between bilingual pairs in SMT, the most popularly used is the IBM model (Brown et al., 1993) in the toolkit GIZA++ (Och et al., 1999), combined with symmetric heuristics.

We estimate the word posterior probability according to word alignments of examples. We create a counter for each word, which might involve in the final translation. The words can come from target parts of examples or the skeleton translation.

For each word, its counter works as follows: 1) Initialize the counter as 0. 2) Keep the counter unchanged if the word either comes from the original translation or does not appear in alignments. 3) Increase the counter by one if its corresponding source word in alignments also appears in the source input sentence. 4) Decrease the counter by one if its corresponding source word in alignments does not disappear in the source input sentence.

Then we estimate posterior probability of the word  $w$  for each fusing translation  $E_i$  by the counter value as following formula:

$$p(w|E_i) = \frac{1}{1 + e^{-c}} \quad (2)$$

where  $c$  is the counter value. The value of  $E_i$  is defined as follows:

$$E_i = \begin{cases} ex_{E_i}, & i = 1, 2, \dots, n \\ E_0, & i = 0 \end{cases} \quad (3)$$

## 2.3 Features for Fusion

In the practice of translation fusion under SMT system combination framework, six common features are used to guide the decoding:

**Language model:** probability from an N-gram language model.

**Word penalty:** penalty depending on the size (in words) of the hypothesis.

**Null-arc penalty:** penalty depending on the number of null-arcs crossed in the confusion network to obtain the hypothesis.

**N-gram agreement:** the value which is equal to the counts of N-gram matches between fusing translations (examples and original translations) and the hypothesis divided by the number of the fusing translations.

**N-gram probability:** a kind of like language model trained on the top-n examples.

**Word confidence:** the production of word posterior probability and the confidence of the fusing translation where the word come from.

The practical effect is that the word posterior probability is computed with a simple method at the cost of estimating the word confidence from original translation roughly. To solve this problem, an original word penalty feature is introduced into our method.

**Original word penalty:** penalty depending on the number of words from the original translation. The feature indicates the degree of repairing original translation.

In addition, although the incremental TER alignment is used in constructing the confusion network to avoid most of alignment errors, overcoming the noise from the examples is critical. So we adopt repetitive word penalty to debilitate this effect.

**Repetitive word penalty:** penalty depending on the number of repetitive words in the hypothesis.

### 3 Experiments

#### 3.1 Experimental Settings

Our experiments are carried out on the HIT dataset in the OLYMPICS task of IWSLT 2012 Evaluation Campaign (Federico et al., 2012). We take the training dataset as examples corpus, which contains 52,603 pairs of Chinese-English sentences. Development and test dataset provided by the task contain 2,057 and 998 pairs of Chinese-English sentences, respectively. The Chinese text is segmented by Stanford Word Segmentation (Chang et al., 2008). Detailed statistics of the corpus are shown in Table 1.

	sent	Segment(zh) Token(en)
Example corpus	52,603	495,638 (zh) 527,599 (en)
Dev	2,057	19,457(zh) 20,782(en)
Test	998	10,047(zh) 11,004(en)

Table 1. The Description of HIT dataset

The original MT outputs of develop set and test set come from Google Translate services. The 5-gram English target language model has been trained on Example corpus using SRILM (Stolcke, 2002). The model parameters are trained by MERT (Och, 2003). The 500-best list is created at each MERT iteration and is appended to the n-best lists created at previous iterations. The results are evaluated by BLEU-4 (Papineni et al., 2002) score.

To grasp the distribution of test set on similar score, the composition of test subsets based on similar scores is calculated, which is shown in Table 2.

	Sent	Segment	Segment/Sent
[0.9,1.0)	36	235	6.53
[0.8,0.9)	209	1,394	6.67
[0.7,0.8)	423	3,218	7.61
[0.6,0.7)	720	6,407	8.90
[0.5,0.6)	931	9,121	9.80
[0.4,0.5)	923	9,462	10.25
(0.0,0.4)	570	5,917	10.38

Table 2. Composition of test subsets based on similar scores

#### 3.2 Evaluating Translation Quality

In our experiments, firstly we re-rank the retrieval examples corpus by the Cosine Similarity score and empirically retrieve the top-15 similar examples for each source sentence in development and test dataset. Secondly, using the Google Translate services to translate the source sentence in test and development set, we obtain the results of its translations (Original). Then we tune the model parameter on the development dataset, and decode on the test dataset to generate new translations (Repaired). For the comparison, we list two results of our baseline method: One is the result of original translation (Original); the other baseline is the result of replacing original translations by the example with max similar score (Replaced). When combining the top-15 similar examples and original translation, the BLEU score of word-level oracle system (Oracle) is shown in Table 3, and the best system (IWSLT12\_Best) on the dataset in IWSLT 2012 Evaluation Campaign is also listed.

As we can see from Table 3, we still obtain significantly inferior results compared to the original translation if we replace all the Google translations by the most similar examples, which is reflected by an absolute 8.55 point drop on the test set in BLEU score. On the other hand, our repairing method, which can repair original

translation result automatically in word-level, leads to an increase of 0.64 absolute BLEU point on the test set.

Model	BLEU%
<i>Original</i>	18.77
<i>Replaced</i>	10.22
<i>Repaired</i>	<b>19.41*</b>
<i>IWSLT12 Best</i>	19.17

Table 3. Comparison with others on BLEU score (\* significant at 0.005-level compared with the score of Original)

The experimental results show that our retrieval examples driven method appears to be effective in repairing incorrect translation with significant improvement in translation quality. Replacing by the most similar example cannot improve the translation quality when the similar score is low. Combing the examples with original translation improves the translation quality. In this sense, it is promising to correct translation by the examples via the proposed method.

### 3.3 The Effect of Example Similarity

We compare our method (Repaired) with two baselines (Original and Replaced) in different similar score region. We evaluate the translations by BLEU score. The results are listed in Table 4.

	<i>Original</i>	<i>Replaced</i>	<i>Repaired</i>
(0.9, 1.0)	17.23	<b>36.99</b>	22.98
(0.8, 0.9]	20.92	<b>27.20</b>	21.44
(0.7, 0.8]	19.90	15.03	<b>20.23*</b>
(0.6, 0.7]	18.55	9.12	17.71
(0.5, 0.6]	18.38	4.22	17.50
(0.4, 0.5]	18.76	1.78	17.54
(0.0, 0.4]	18.25	0.80	17.20

Table 4. BLEU in different similar score region (\*significant at 0.005-level compared with the score of Original)

From Table 4, we can see when the similar score is greater than 0.8, replacing the original translation by the most similar example has a serious advantage on BLEU score. When the similar score declines, the BLEU score also drops sharply. When the similar score region is (0.7, 0.8], our method has a significant improvement of absolute 0.33 BLEU score compared with original. And when the similar score declines bellow 0.7, the original translation is better. But it is remarkable that the result is generated by un-tuned parameters model.

### 3.4 Feature Analysis

In the experiment, we investigated the contribution of our different feature sets. After removing one feature, we retune the weights of features on the development set and re-decode on the test set. We evaluate the outputs of these models by BLEU, and list the results in Table 5.

Model	BLEU%
<i>Repaired</i>	19.41
Without word penalty	19.39
Without N-gram agreement	19.33
Without language model	19.23 <sup>^</sup>
Without repetitive word penalty	19.21 <sup>^</sup>
Without null-arc penalty	19.10 <sup>^</sup>
Without original word penalty	19.01 <sup>^</sup>
Without word confidence	18.98 <sup>^</sup>
Without N-gram probability	18.71 <sup>^</sup>

Table 5. Contribution of Features (^significant at 0.05-level compared with the repaired score)

As shown in Table 5, the performance drops significantly ( $p < 0.05$ ) when language model, repetitive word penalty, null-arc penalty, original word penalty, word confidence, N-gram probability is removed from the feature set respectively. And word penalty and N-gram agreement have weak effects on the results. It is remarkable that our specific features repetitive word penalty, original word penalty, and word confidence can bring the improvement of 0.20, 0.40, and 0.43 BLEU point than that without them respectively.

## 4 Conclusion

In this paper, we introduce statistical confusion network for translation example fusion to improve the current online MT quality. We estimate the posterior of the word translation by the example similarity and introduce some new features to enhance the log-linear model optimization for the best translation. We check our method on the HIT dataset in the OLYMPICS task of IWSLT 2012 Evaluation Campaign. The Experimental results indicate that proposed method enhance the Chinese-English translations by Google, with a significant 0.64 absolute improvement according to BLEU score.

### Acknowledgments

This work is supported by the NSF China (No. 61272384 & 61105072) and the National High Technology Research and Development Program of China (863 Program, No. 2011AA01A207).

## References

- Biçici Ergun and Marc Dymetman. 2008. Dynamic translation memory: using statistical machine translation to improve translation memory fuzzy matches. In Proceedings of the 9th international conference on Computational linguistics and intelligent text processing, pages 454-465.
- Brown, Peter F., Stephen A. Della-Pietra, Vincent J. Della-Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation. *Computational Linguistics*, 19(2): 263–313.
- Chang Pi-Chuan, 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.
- Daumé III Hal and Jagadeesh Jagarlamudi. 2011. Domain Adaptation for Machine Translation by Mining Unseen Words. In *Proceedings of ACL11-HLT*, pages 407-412.
- He Yifan, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with translation recommendation. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 622-630.
- Och Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL03*, pages 160-167.
- Och Franz Josef, Christoph Tillman, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC), pages 20–28.
- Koehn Philipp and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.
- Ma Yanjun, Yifan He, Andy Way, and Josef van Genabith. 2011. Consistent Translation using Discriminative Learning - A Translation Memory-inspired Approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1239-1248.
- Simard Michel and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. The Twelfth Machine Translation Summit, pages 120-127.
- Smith James and Stephen Clark. 2009. EBMT for SMT: a new EBMT--SMT hybrid. In *Proceedings of the 3rd International Workshop on Example-Based Machine Translation*, pages 3-10.
- Stolcke Andreas. 2002. SRILM - An Extensible Language Modeling Toolkit, in Proc. Intl. Conf. Spoken Language Processing.
- Papineni Kishore, 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.



# Phrase-based Parallel Fragments Extraction from Comparable Corpora

Xiaoyin Fu, Wei Wei, Shixiang Lu, Zhenbiao Chen and Bo Xu

Interactive Digital Media Technology Research Center, Institute of Automation,  
Chinese Academy of Sciences, Beijing, China

{xiaoyin.fu, wei.wei, shixiang.lu, zhenbiao.chen, xubo}@ia.ac.cn

## Abstract

We present a phrase-based method to extract parallel fragments from the comparable corpora. We do this by introducing a force decoder based on the hierarchical phrase-based (HPB) translation model to detect the alignments in comparable sentence pairs. This method enables us to extract useful training data for statistical machine translation (SMT) system. We evaluate our method by fragment detection and large-scale translation tasks, which show that our method can effectively extract parallel fragments and improve the performance of the state-of-the-art SMT system.

## 1 Introduction

Parallel corpora are valuable resources for training a statistical translation system. In most cases, it has been an effective way to build state-of-the-art statistical models using a large scale of parallel corpora. However, the parallel corpora only exist in particular domains for a few number of language pairs, such as international conference recordings and legal texts. Since comparable corpora exist in large quantities with many languages, and the exploitation in them for extracting parallel data can be very useful for SMT system, the acquisition of parallel data from comparable corpora has caught much attention.

Various methods (Zhao and Vogel, 2002; Munteanu and Marcu, 2005; Abdul-Rauf and Schwenk, 2009; Smith et al., 2010) have been previously proposed to extract parallel data from comparable corpora at the sentence level. These methods share the same framework, which firstly identifies candidate document pairs and then extracts parallel sentences from the obtained documents. However, it is found that most of these sentences are comparable sentence pairs (Hong et

al., 2010), which embed non-parallel fragments or even lack translations. Consider the comparable sentence pair from Chinese to English in Figure 1. Methods for extracting parallel sentences will bring in noise when these bilingual sentences are retained. But discarding them is also not a wise choice, as there are still some useful parallel fragments as the underlines shown in the figure.

中国 政府 开始 大力 发展 中国 的 经济。

And developing the economy of China is the practical choice for the Chinese government.

Figure 1: Example of comparable sentence pairs. The parallel fragments are marked by underlines.

In order to deal with this problem, further efforts (Munteanu and Marcu, 2006; Quirk et al., 2007; Kumano et al., 2007; Lardilleux et al., 2012) were made to obtain parallel data at the fragment level. The work of (Riesa and Marcu, 2012) detected parallel fragments using the hierarchical alignment model. However, this approach obtains fragments from parallel sentence pairs, which limits its application in comparable corpora. (Hewavitharana and Vogel, 2011) have explored several alignment approaches to detect parallel fragments embedded in comparable sentences. However, these approaches extract fragments mainly using the lexical features and considering the words in parallel fragments are independent, which make it difficult to measure the alignments exactly.

In this paper, we present a phrase-based method, which considers both the lexical and phrasal features, to extract parallel fragments from comparable corpora. We introduce a force decoder based on the HPB translation model to detect parallel fragments for each sentence pair. The results show that our method can effectively extract parallel fragments from the comparable corpora.

ra and significantly improve the performance on Chinese-to-English translation tasks.

## 2 Parallel Fragments Extraction

### 2.1 HPB Translation Model

The HPB translation model (Chiang, 2005) has shown strong abilities in SMT for its capability in generalization. It is based on the weighted synchronous context-free grammar (SCFG). And the translation rule is represented as:

$$X \rightarrow \langle \alpha, \gamma, \sim \rangle \quad (1)$$

where  $X$  is a non-terminal,  $\alpha$  and  $\gamma$  are source and target strings with terminals and non-terminals.  $\sim$  describes a correspondence between the non-terminals in  $\alpha$  and  $\gamma$ .

Two glue rules are added so that it prefers combining hierarchical phrases in a serial manner:

$$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle \quad (2)$$

$$S \rightarrow \langle X_1, X_1 \rangle \quad (3)$$

### 2.2 Force Decoding based on HPB model

The force decoding can be seen as a bilingual parsing process that generates derivation trees from both sides of the sentence pair with an existing HPB model.

Let  $\mathbf{e} = e_1^M$  and  $\mathbf{f} = f_1^N$  be the source and target sentences in comparable corpora. For each of the sentence pair  $\mathbf{e}$  and  $\mathbf{f}$ , the decoding process enumerates all of the possible bilingual derivation trees  $\Phi$  with HPB rules from bottom to top. At each node in these derivation trees, the decoder generates alignments by recursively combining phrases generated from the current node's children, and builds up larger and larger alignments. It should be noted that these nodes can be generated only if the alignments are exactly contained in both elements of the sentence pair. The derivation process works similarly to a CKY parser, moving bottom-up and generating larger constituents. However, the force decoder generates derivation trees for both of the bilingual sentences simultaneously and these trees do not have to span the entire sentences, especially in the non-parallel sentences, which is quite different with the CKY parser.

Still considering the comparable sentence pair in Figure 1. Figure 2 gives an example of extracting one of the parallel fragments by force decoding with the following HPB rules:

$$X \rightarrow \langle \text{发展 } X_1, \text{developing } X_1 \rangle$$

$$X \rightarrow \langle X_1 \text{ 中国}, \text{China} \rangle$$

$$X \rightarrow \langle X_1 \text{ 经济}, \text{the economy} \rangle$$

$$X \rightarrow \langle X_1 \text{ 的 } X_2, X_2 \text{ of } X_1 \rangle$$

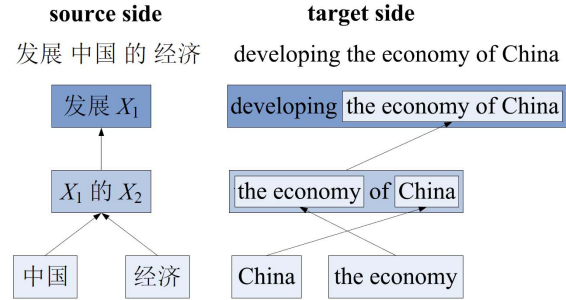


Figure 2: Example of derivation trees in force decoding. To give better illustration, the non-terminal rules from parent nodes are combined with the rules from child nodes on the target side.

It can be seen that the bilingual derivation trees, which represent the source and target fragments, are generated simultaneously. At the first derivation step, it is found that the Chinese words “中国” and “经济” from the source side can be translated into English as “China” and “the economy”, which are exactly contained in the target sentence. Then we keep these words as the nodes in bilingual derivation trees, and continue to generate parent nodes by combining these child nodes bottom-up. The derivation process continues until there are no bilingual nodes that meet the words in both of bilingual sentences. At last, we will get the parallel fragments “发展中国的经济” and “developing the economy of China” from the top of the derivation trees.

### 2.3 The Extension of HPB Rules

In our force decoding framework, there are some words that do not have translation rules, such as the out-of-vocabulary (OOV) words. This case could make up a large portion in the comparable corpora. To overcome this drawback, the HPB model has to be trained on a large scale of training data with a large vocabulary. Even so, there are still some of the words that may lack translations. We suppose these words can be translated into any of the sequential words in target sentences and add a special rule to our HPB model:

$$X \rightarrow \langle e_i, f_{(i', j')} \rangle, 1 \leq j' - i' \leq 2 \quad (4)$$

where  $i$  is the position of the word that do not have translation rules in source side,  $i'$  and  $j'$  are the start and end positions of the phrasal segment in target sentence. Here we restrict the length of the phrasal segment because larger segment tend to bring in noise in force decoding.

Moreover, in order to better evaluate the alignments between the parallel fragments, we extend the original HPB rules inspired by the work of (Čmejrek et al., 2009):

$$\langle X_1, X_1 f \rangle, \langle X_1, f X_1 \rangle, \langle X_1 e, X_1 \rangle, \langle e X_1, X_1 \rangle \quad (5)$$

$$\langle X_1 X_2, X_2 X_1 \rangle \quad (6)$$

in which rules (5) allow the HPB rules to insert and delete a single word, and rule (6) expands the standard glue rules and enables the aligning phrasal segments swap their constituents.

## 2.4 The Verification of Parallel Fragments

For each bilingual sentence pair, we can generate various alignment derivation trees. The derivation trees from source side are isomorphic to the target side because of the characteristic of SCFG.

In order to better evaluate the alignment for the derivation trees, each HPB rule in force decoding is associated with a score that is computed via the following log linear formula:

$$w(X \rightarrow \langle \alpha, \gamma, \sim \rangle) = \prod_i \phi_i(f, e)^{\lambda_i} \quad (7)$$

where  $\phi_i(f, e)$  is a feature describing one particular aspect of the rule associated with the source and target phrases ( $f, e$ ), and  $\lambda_i$  is the corresponding weight of the feature. Following the standard HPB model, features used in our force decoding are relative-frequency phrase translation probability  $P(f|e)$  and its inverse  $P(e|f)$ , lexically weighted phrase translation probability  $lex(f|e)$  and its inverse  $lex(e|f)$ .

Moreover, we consider the score of the special rule is:

$$w(X \rightarrow \langle e_i, f_{i',j'} \rangle) = \omega \times e^{-|j'-i'|} \quad (8)$$

in which,  $\omega$  is the weight of the special rule.

After generating the derivation trees, we recursively traverse these trees at each node top-down, and extract parallel fragments from both sides with the following constraints:

1) The node in the derivation tree has a score greater than a threshold  $\tau$ .

2) The node that represents the words from source side whose span is greater than 2.

The first constraint forces us to extract fragments with high alignment scores, as there are some alignment errors in HPB rules. And the second constraint makes us be more confident in the alignment scores over the larger fragments. The recursive traversal from derivation trees stops, once a fragment pair has been extracted.

For each sentence pair, different parallel fragments are extracted from derivation trees. Then we combine these fragments if there are overlaps in both source and target side. Otherwise, we keep these fragments as independent pairs.

## 3 Experiments

In our experiments, we compared our fragments extraction method with the PESA method explored by (Hewavitharana and Vogel, 2011), which is based on the lexical features.

### 3.1 Data and Evaluation Setup

We used the parallel corpora from LDC<sup>1</sup> to train our HPB model in force decoding. The HPB model was trained following (Chiang, 2007) with word alignment by running GIZA++ (Och and Ney, 2003). We downloaded comparable data from the online news sites: the BBC, and Xinhua News. The candidate sentence pairs (**Raw**) had been extracted following the approach of (Munteanu and Marcu, 2005) as we only focused on the performance of parallel fragments extraction. The sizes of these corpora are listed in Table 1.

Data Sets	#Sentences	#Chinese	#English
LDC	3.4M	64M	70M
Raw	2.6M	42M	49M

Table 1: Numbers of sentences and words for the parallel and comparable corpora.

We evaluated the quality of the extracted parallel fragments in two different ways:

**Fragments Evaluation** We obtained manual alignments for 600 sentence pairs and extracted parallel segments up to 10 words that are consistent with the annotated word alignment. We also removed the segments less than 3 words for the

<sup>1</sup>LDC2002E18, LDC2002T01, LDC2003E07, LDC2003E14, LDC2003T17, LDC2004T07, LDC2004T08, LDC2005T06, LDC2005T10, LDC2005T34, LDC2006T04, LDC2007T09.

constraint as described in Section 2. Then we tested the performance with the manual annotation.

**Translation Evaluation** We evaluated the fragments on Chinese-to-English translation tasks. We used a HPB translation system with a 4-gram language model trained on about 4 billion words of English using SRI Language Toolkit (Stolcke, 2002). We tuned parameters of the SMT system using minimum error-rate training (Och, 2003) to maximize the BLEU-4 (Papineni et al., 2002) on NIST 2005, and evaluated on the standard test sets, NIST 2006 and NIST 2008.

## 3.2 Experimental results

### 3.2.1 Performance on Fragments Extraction

We first compared the our method (HPB-FD) with PESA by fragments extraction. To give credit to our fragments extraction, we used partial matches to evaluate the performance of our extract method, following the way of (Hewavitharana and Vogel, 2011). The precision and recall were defined based on the tokens in the extracted target fragments that were also exists in the reference. And the F1 score was calculated in the standard way.

	<b>Exact</b>	<b>P</b>	<b>R</b>	<b>F1</b>
PESA	60.36	88.42	84.74	86.54
HPB-FD	74.12	94.36	88.90	91.55

Table 2: The results for fragments extraction with PESA and HPB-FD.

Table 2 gives the performance of PESA and our HPB-FD method. The results are presented as percentages of: exact matches found (**Exact**), precision (**P**), recall (**R**) and **F1**. It can be seen that our method can effectively extract parallel fragments from the comparable corpora. Comparing to PESA, our extraction method has higher scores in both Exact and F1 measure. This demonstrates that extracting fragments by our force decoding method can be more effectively to evaluate parallel fragments in comparable corpora.

### 3.2.2 Performance on Machine Translation

We then evaluated the extracted parallel fragments with the HPB translation system. In the baseline system, translation model (LDC) was trained on the LDC corpora that had been cleaned and thought to be less noisy. In the contrast experiments, we trained three translation models. The first model (LDC+Raw) was trained on the LDC

with the extracted comparable sentences. The second model (LDC+PESA) was trained on the LDC and fragments that were extracted by PESA. And the third (LDC+HPB-FD) was trained on LDC and fragments that were extracted by HPB-FD. Table 3 lists the BLEU scores obtained by different training data.

	<b>NIST 2006</b>	<b>NIST 2008</b>
LDC	28.07	26.12
LDC+Raw	28.20(+0.13)	26.05(-0.07)
LDC+PESA	28.65(+0.58)	26.62(+0.50)
LDC+HPB-FD	<b>29.01(+0.94)</b>	<b>26.93(+0.81)</b>

Table 3: The translation performance with different training data. BLEU score gains are significant with  $p < 0.01$ .

Comparing to the baseline system, all the adding training data get stable improvements in translation performance except for the comparable sentences. It suggests that the simple increment in training data does not always lead to better performance. The superiority of parallel corpora confirms that, the quality is more important than quantity in collecting training data. Moreover, comparing to the parallel fragments extracted by PESA, our method get better translation results in both translation tasks, which also suggests our method can effectively extract parallel fragments from comparable corpora for the SMT system.

## 4 Conclusions

Parallel data in the real world is increasing continually. However, we cannot always get the translation performance improved by simply enlarging our training data. The collection of parallel data is expensive, and to our best knowledge, there is not a unified method to detect parallel fragments automatically.

We have presented an effective phrase-based method, which combines the lexical and phrasal features, for extracting parallel fragments from comparable corpora. The similarity between the source and target fragments is measured by the force decoding based on the existing HPB model. Experimental results show that our method can effectively detect the parallel fragments and achieve significant improvements over the baseline HPB translation system on the large scale Chinese-to-English translation tasks.

## Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National High Technology Research and Development Program ("863" Program) of China under Grant No.2011AA01A207 and also supported by the National Program on Key Basic Research Project ("973" Program) under Grant No.2013CB329302.

## References

- Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the Use of Comparable Corpora to Improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 16–23.
- David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270.
- David Chiang. 2007. Hierarchical Phrase-based Translation. *Computational Linguistics*, 33(2):201–228.
- Martin Čmejrek, Bowen Zhou, Bing Xiang. 2009. Enriching SCFG Rules Directly from Efficient Bilingual Chart Parsing. In *Proceedings of the International Workshop on Spoken language Translation*, pages 136–143.
- Sanjika Hewavitharana and Stephan Vogel. 2011. Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 61–68.
- Gumwon Hong, Chi-Ho Li, Ming Zhou and, Hae-Chang Rim. 2010. An Empirical Study on Web Mining of Parallel Data. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 474–482.
- Tadashi Kumano, Hideki Tanaka and Takenobu Tokunaga. 2007. Extracting Phrasal Alignments from Comparable Corpora by Using Joint Probability SMT Model. In *Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 95–103.
- Adrien Lardilleux, François Yvon and Yves Lepage. 2012. Hierarchical Sub-sentential Alignment with Anymalign. In *Proceedings of the 16th annual meeting of the European Association for Machine Translation*, pages 279–286.
- Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting Parallel Sub-Sentential Fragments from Non-Parallel Corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 81–88.
- Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 160–167.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.
- Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the ACL*, pages 311–318.
- Chris Quirk, Raghavendra U. Udupa, and Arul Menezes. 2007. Generative Models of Noisy Translations with Applications to Parallel Fragment Extraction. In *Proceedings of the Machine Translation Summit XI*, pages 377–384.
- Jason Riesa and Daniel Marcu. 2012. Automatic Parallel Fragment Extraction from Noisy Data. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 538–542.
- Jason R. Smith and Chris Quirk and Kristina Toutanova. 2010. Extracting Parallel Sentences from Comparable Corpora using Document Level Alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 403–411.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving Machine Translation Performance by Exploiting Non-parallel Corpora. *Computational Linguistics*, 31(4).
- Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of International Conference on Spoken Language Processing*, pages 901–904.
- Bing Zhao and Stephan Vogel. 2002. Adaptive Parallel Sentences Mining from Web Bilingual News Collection. In *2002 IEEE Int. Conf. on Data Mining*, pages 745–748.

# A Hybrid Approach for Anaphora Resolution in Hindi

**Praveen Dakwale**  
LTRC, IIIT-Hyderabad  
India

dakwale.praveen@gmail.com

**Vandan Mujadia**  
CSPIT, Charusat  
Gujarat, India

vmujadia@gmail.com

**Dipti M Sharma**  
LTRC, IIIT-Hyderabad  
India

diptims@gmail.com

## Abstract

In this paper we present a hybrid approach to resolve Entity-pronoun references in Hindi. While most of the existing approaches, syntactic as well as data-driven, use phrase-structure syntax for anaphora resolution, we explore use of dependency structures as a source of syntactic information. In our approach, dependency structures are used by a rule-based module to resolve simple anaphoric references, while a decision tree classifier is used to resolve more ambiguous instances, using grammatical and semantic features. Our results show that, use of dependency structures provides syntactic knowledge which helps to resolve some specific types of references. Semantic information such as animacy and Named Entity categories further helps to improve the resolution accuracy.

## 1 Introduction

In various approaches on anaphora resolution syntax has been used as an important feature. Some well-known syntax based approaches include Hobbs algorithm (Hobbs, 1986) and the Centering approach (Brennan et al., 1987). Various rule based and data driven approaches have been proposed which use syntactic information as an important feature.

Most of the earlier works have used phrase-structure parse as a source of syntactic information. However, dependency structures are more suitable representations for relatively free word order languages such as Hindi (Bharati et al., 1995; Melčuk, 1988) and hence research in many such languages has focused on development of dependency based resources resulting in better availability of dependency data for such languages. In this paper, we explore the possibility of using dependency structure for anaphora resolution in Hindi.

However, we do not intend either to propose dependency as an alternative to phrase structure or to compare the usability of the two frameworks.

(Prasad and Strube, 2000) is one of the most important approach for anaphora resolution in Hindi. They applied a discourse salience ranking to two pronoun resolution algorithms, the BFP and the S-List algorithm. (Dakwale and Sharma, 2011) reported the best performance for Hindi in Anaphora Resolution tool contest in Indian languages(ICON-2011). They propose a hybrid approach with limited linguistic knowledge such as NER categories and verb similarity.

Two earlier approaches explore the use of dependency relations for anaphora resolution. For Hindi, (Uppalapu and Sharma, 2009) extends the S-List algorithm by using two different lists in place of a single list. For English, (Björkelund and Kuhn, 2012) explores the possibility of using dependency relations as a feature for co-reference resolution in a learning based approach. Both the approaches are limited in their exploration of dependency for anaphora resolution as they only use dependency relations either as a salience for ranking the candidate referents or as an additional feature in a learning based approach. We discuss how resolution of different types of Entity-pronoun references in Hindi can benefit from dependency relations and semantic information. We present a hybrid approach to resolve Entity-pronoun references in Hindi which combines a rule-based system that uses dependency structures and relations and further improvement is achieved with semantic information such as animacy.

## 2 Data set and Grammatical Framework

In this work we use the data from the ‘Hindi/Urdu Dependency Treebank’ (Bhatt et al., 2009). It is a rich corpus with various linguistic information. The dependency annotation in this treebank is based on the Computational Paninian Grammar

(CPG-henceforth) framework, as is explained in (Begum et al., 2008) and (Bharati et al., 1995). This framework is based on the notion of ‘*karaka*’ which are syntactio-semantic relations representing the participant elements in the action specified by the verb and it emphasizes the role of case endings or markers such as post-positions and verbal inflections.<sup>1</sup> Table 1 shows some of the relevant relations and their rough correspondence to the traditional grammatical relations in English.

Label	CPG relation	Grammatical/thematic equivalent
k1	<i>karta</i>	Subject
k2	<i>karma</i>	Object
k4	<i>sampradan</i>	Experiencer/reciever
k7p(or k2p)	<i>adhikaran</i>	Location
r6	<i>sambandh</i>	Genitive

Table 1: CPG relations and equivalents

We use a part of the treebank which is also annotated with animacy information for Noun phrases as described in (Jena et al., 2013). Also, we used NE-Recognizer for Hindi to get the Named entity categories. The treebank has been annotated with anaphora links for all the pronouns as per the scheme described in (Dakwale et al., 2012). The size of the data that we use for our experiments is 325 documents, containing 4970 pronouns out of which 3233 pronouns are annotated as entity pronouns

### 3 System Description

The hybrid approach in our system is different from other hybrid approaches, in the way that instead of a rule based filtering followed by instance classification, our system includes a rule-based resolution module followed by a decision tree classifier for the remaining unresolved pronouns.

Anaphoric reference type can be classified into abstract (event) references where an anaphora refers to an event or a proposition and concrete (entity) references where it refers to a concrete entity like noun phrase (person,place etc), quantifiers etc. In this work, we focus on resolving only entity pronouns, hence the mention detection or anaphoricity determination step is not required for our system. Certain pronominal forms based on their different syntactic behaviour, can be resolved quite successfully with some specific rules using the dependency information. Therefore, we categorize pronominal forms in four types: Reflexive,

<sup>1</sup>The detailed description of these relations is given in Hindi Dependency tagset (<http://ltrc.iiit.ac.in/MachineTrans/research/tb/dep-tagset.pdf>)

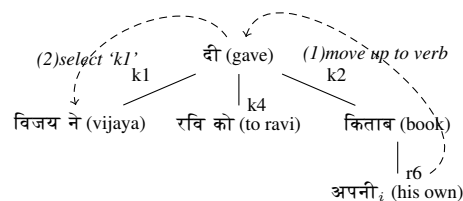


Figure 1

Locative, Relative and Personal pronouns. The pronouns which are identified as concrete in the data are passed to the rule-based resolution module in which different rules depending on the category of the pronoun are applied to identify the correct referent. If none of the possible rules apply to a pronoun, it is passed to the classifier which uses a learning algorithm to identify the referent.

### 3.1 Rule based resolution module

The rule based module attempts to resolve the pronoun, using the dependency relations and other information, based on the category of the pronoun, which is decided using an exhaustive list of pronoun categories. We describe below some of the important rules used for different pronoun categories.

#### 3.1.1 Reflexives

In Hindi *Possessive reflexives* are the most frequent reflexives which are only used in possession relation within the same clause and are different from third person possessive pronouns. Unlike English reflexives, they are not inflected with the gender and number of the possessor, but that of the possession. They include [अपना (*apana*), अपनी (*apanii*), अपने (*apane*)] (own). There are *Non-possessive reflexives* which can be used in any participant position, but mostly used in object position. They include [अपने आप(*apane-aap*), स्वयम्(*swayam*), खुद(*khud*)] representing ‘one-self’. As it can be well derived from the binding theory, the referent of the reflexive pronoun is the accessible subject in its own governing category. Also, the ‘k1’ relation of CPG-based framework roughly corresponds to ‘SUBJECT’ of the traditional framework, thus the referent of the reflexive pronoun in most cases is the ‘k1’ of the same clause, i.e. that child node of the root verb of the clause, which has a dependency relation ‘k1’.

- (1) विजय<sub>i</sub> ने रवि को अपनी<sub>i</sub> किताब दी  
vijay.ERG ravi.DAT his.POSS.REF book gave  
‘vijay<sub>i</sub> gave (his own)<sub>i</sub> (POSS.REF) book to ravi.’

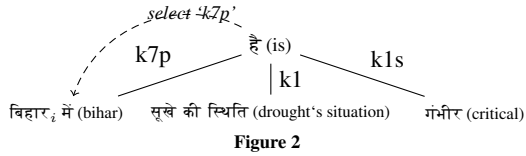


Figure (1) shows the dependency structure of example (1). The root verb of the clause containing possessive reflexive अपनी (*his*) is दी (*gave*) which has a descendant node विजय (*vijay*) with a dependency relation ‘k1’ with the verb. Thus it should be selected as the referent of the pronoun.

### 3.1.2 Place pronouns

Locative pronouns refer to location or places. They include वहाँ (‘there’) and यहाँ (‘here’). In CPG-based framework (as discussed in section 2), separate labels are used to represent the locative case, thus it can help in identifying the referents of these pronouns. To resolve place pronouns, we use dependency relations and Named entity Categories. Thus, place pronoun can be resolved by selecting the noun phrase nearest to the pronoun which has ‘LOCATION’ as NER-Category or the nearest NP with the dependency label ‘k7p’ or ‘k2p’. For ex :

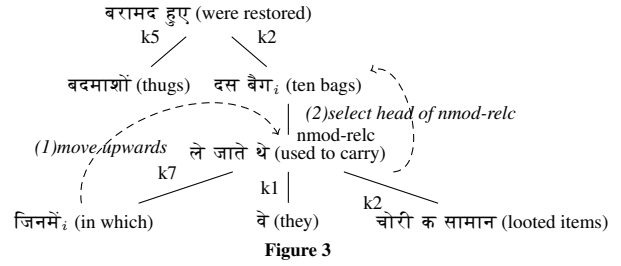
- (2)  $[\text{NER}=\text{LOC}]$  बिहार में सूखे की स्थिति गंभीर है। आज  
 bihar.LOC drought's situation critical is. today  
 प्रधानमंत्री ने वहाँ का दौरा किया।  
 prime minister there visited  
 ‘Situation of drought is critical in bihar. Today Prime minister visited there.’

Figure (2) shows the dependency structure of example(2). Noun phrase with NER category as ‘LOCATION’ nearest to the pronoun वहाँ (*there*) is (*bihaara*), thus it can be selected as the referent. In absence of NE category, dependency relations can be used to identify the referent.

### 3.1.3 Relative pronouns

In Hindi, relative pronouns include जो (which) and its case forms such as जिसे (to which), जिससे (from which) etc. In the CPG-based framework, relative clauses are marked with a relation ‘nmod-relc’, i.e. the relative clause is attached below that noun phrase which is relativized by the clause and the relation is labeled as ‘nmod-relc’. Thus, the referent of the relative pronoun should be selected as the noun-phrase to which the clause containing relative pronoun is attached. Consider following example

- (3) बदमाशों से दस बैग बरामद हुए जिनमें वे चोरी का  
 From thugs ten bags restored in which they looted  
 सामान ले जाते थे  
 items used to carry

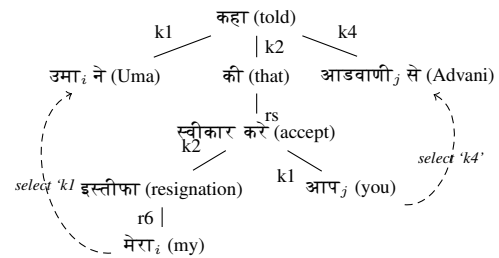


‘Ten bags<sub>i</sub> were restored from the thugs in which<sub>i</sub> they used to carry the looted items.’

Figure (3) shows the dependency structure of example (3), in which the relative pronoun is (‘which’) and the head of the relative clause is the verb ‘ले जाते थे’ (*used to carry*) which in turn is attached below the NP node (‘ten bags’) with a relation ‘nmod-relc’, which is selected as the referent of the relative pronoun.

### 3.1.4 Personal pronouns

All personal pronouns in Hindi are marked for number, respect and case. We consider resolution of first and second person pronouns separate from third person pronouns.



In the news corpus data, first and second person pronouns mostly occur in the narrative or attributional clauses (those subordinate clauses whose main clause has an attribution root verb such as बोल (*to tell*), कह (*to say*), बता (*to tell*) etc). If the first person pronoun is a part of attributional clause then its reference is the speaker of that clause. It is almost always ‘k1’ of the main clause. Similarly the referent of a second person pronoun in an attributional clause, is mostly the ‘k4’ or experiencer of the main clause. For ex :

- (4) उमा ने आडवाणी से कहा की आप  
 Umaa.ERG advaani.DAT said that you.HONORIFIC.ACC  
 मेरा इस्तीफा स्वीकार करे  
 my resignation accept  
 ‘Umaa<sub>i</sub> said to advaani<sub>j</sub> that you<sub>j</sub> accept my<sub>i</sub> resignation.’

Figure (4) shows the dependency structure of example (4), in which (*you*) is second person pronoun in the attributional clause rooted at (*accept*), hence its referent is selected as the ‘k4’ of the main clause i.e. (*advani*). Similarly for the first person



pronoun (*my*), referent is selected as ‘k1’ of the main clause i.e. (*Umaa*).

References of third person pronouns mostly are inter-clausal or inter-sentential. For third person anaphora, we adopt re-ordering of the candidate elements based on the salience of dependency relations, similar to (Uppalapu and Sharma, 2009). However, with two modifications : First, they consider the salience ordering ( $k1 > k2 > k3 > k4 > others$ ) to rank the candidate elements, similar to ordering of the grammatical relation (*subject > direct object > indirect object > adjunct*) as in (Prasad and Strube, 2000). We adopt a slightly modified ordering of the relations ( $k1 > k2 > r6 > k4 > k3 > others$ ) based on the relative frequency of the dependency relations for animate entities. Second, we also use animacy along with number to prune the candidate NP list. For ex :

- (5)  $[_{k1,h}$  उमा $_i$  ने $]$   $[_{k7}$  पहले $]$   $[_{k4,h}$  शिवराज $_j$  को $]$   $[_{k2,rest}$  पत्र $]$   
 Uma.ACC first shivraaj.ACC letter  
 लिखा। फिर  $[_{k1,h}$  उन्होंने $_i$   $[_{k4,h}$  उन्हें $_j$   $[_{k3,rest}$  कोर्ट से $]$   
 wrote. later she.HON.ACC him.HON from court  
 $[_{k2,rest}$  नोटिस $]$  भिजवाया  
 notice sent.CAUSATIVE

‘First uma<sub>i</sub> wrote a letter to shivraaj<sub>j</sub>. Later she<sub>i</sub> sent a court notice to him<sub>j</sub>’

In the above example there are two pronouns in the second sentence : first is उन्होंने<sub>i</sub> (*she*) (gender neutral). Salience based ordering of the possible referents for this pronouns is : [(*umaa*), पत्र (*letter*), (*shivraaj*)]. Since the top element i.e. (*umaa*) agrees with the pronoun in number and animacy, it is selected as the referent for (*she.ACC*). Thus the ordered list of candidates becomes : [पत्र (*letter*), (*shivraaj*)]. The second pronoun is (*him*) (gender neutral), but the top element in the list (*letter*) doesn’t agree with pronoun either in number or animacy. However, the next element in the list (*shivraaj*) agrees with the pronoun for both features, hence it is selected as the referent of the pronoun. If a pronoun could not be resolved within the two sentence, it is passed for learning based resolution.

### 3.2 Classifier module

We use the approach of (Soon et al., 2001) for classification. For training, a positive instance is created by pairing each anaphora and its actual antecedent, and negative instances are created by pairing the anaphora with multiple preceding non-antecedent Noun phrases. For testing, unlabeled instances are created by pairing the anaphora with all the Noun-phrases in upto 3 previous sentences. Testing instances are classified as positive

or negative based on the model learned in the training phase. Positively labeled instances are then re-ranked based on the decision-tree-confidence-factor as described in (Witten and Frank, 2005). The NP-candidate corresponding to the highest ranked instance is proposed as the referent of the pronoun.

#### 3.2.1 Features

Following features are used for classification:

- Number : singular, plural, honorific
- Named Entity categories: ‘Person’, ‘Organization’, ‘Location’, ‘Number’
- Distance feature: #NP chunks and #sentences between the pronoun and the candidate NP.
- Animacy : ‘human’, ‘animate’, ‘rest’.

## 4 Evaluation

We divide the treebank data approximately into ratio 2:1 for training and testing respectively. The training data contains 2162 entity pronouns and the test data contains 1071 entity pronouns.

### 4.1 Results

Table (2) shows the accuracies for different types of pronouns resolved by the rule-based module.

	Total pronouns	Correct Resolved	Accuracy
Reflexive Pronoun	156	129	.82
Relative Pronouns	80	68	.85
Locative Pronouns	48	37	.77
1st and 2nd person Pronouns	81	76	.93
Third person Pronouns	706	343	.48
Overall (Rule based system)	<b>1071</b>	<b>653</b>	<b>.60</b>

Table 2: Accuracy of the rule-based system

Results in Table 2 show that performance of the rule based system is quite high for all types of pronouns except for third person personal pronouns. This motivates us to use a learning based approach for the pronouns which remain unresolved in the first module. Table (3) shows the overall performance of the hybrid system achieved over the rule-based system by using different sets of features. The best performance (**0.70**) is achieved with a combination of all the features.

	Total	Correct	Accuracy
Rule based system(RB)	1071	653	.60
RB+Distance	1071	696	.64
RB+Distance+Agreement	1071	713	.66
RB+Distance+Animacy	1071	731	.68
RB+Dist+Animacy+Agreement	<b>1071</b>	<b>753</b>	<b>.70</b>

Table 3: Accuracy of the hybrid system

We provide a tentative comparison of our approach with two earlier systems : (Dakwale and Sharma, 2011) and (Uppalapu and Sharma, 2009).

Though an exact comparison is not possible due to unavailability of the data used in those systems.

	Total	Correct	Accuracy
Uppalapu-S	142	123	.86
Uppalapu-L	100	64	.64
(Dakwale and Sharma, 2011)	258	134	.52
Our system	<b>1071</b>	<b>753</b>	<b>.70</b>

**Table 4:** Resolution results of the three systems. Uppalapu-S and Uppalapu-L are the results of (Uppalapu and Sharma, 2009) for short and long story data respectively

## 4.2 Discussion and Error analysis

Table (4) shows that our system has achieved noticeable improvement over (Dakwale and Sharma, 2011), which is a knowledge poor approach using limited information. However, our system uses treebank data with information such as dependency and animacy.

(Uppalapu and Sharma, 2009) presents results for two sets of data, i.e. long and short stories. The overall accuracy of our approach is better than the accuracy for their long story data, although it is lower than theirs for their short story data. We have presented our results on treebank data which contains news articles from various domains with average size of 20 sentences, above results show that our approach performs consistently better even for longer texts and domain independent data. Also, the performance of the system for third person pronouns is relatively lower than that of other types of pronouns. Table 5 shows a break-up of the distribution of third person pronouns into two forms: Proximal and Distal, and their accuracies. The accuracy for resolution of proximal pronouns is exceptionally low than that of distal pronouns which can be attributed to the ambiguity in the resolution of distal pronouns which can refer to animate as well as inanimate objects

	Total	Correct	Accuracy
Proximal	132	43	.32
Distal	574	394	.68
Total Third person	706	437	.61

**Table 5:** Separate results for proximal and distal third person

## 5 Conclusion and Future Work

The rule based system achieved a substantial accuracy of 60% which implies that dependency relations can help achieve an acceptable resolution performance for Hindi, and the use of decision tree classifier demonstrated a substantial improvement of 10% over the rule based system's accuracy. This shows that semantic features like animacy and Named entity categories provide important linguistic information for anaphora resolution.

In the current work, we have focused only on the resolution of entity pronoun references. In fu-

ture we aim at the identification of reference type and resolution of event anaphora. We also aim to conduct experiments with dependency structures for anaphora resolution in other Indian languages such as Telugu, Bengali etc.

## References

- Rafiya Begum, Samar Husain, Arun Dhvaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of IJCNLP*.
- Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural language processing: a Paninian perspective*. Prentice-Hall of India.
- Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the LAWIII. ACL*.
- Anders Björkelund and Jonas Kuhn. 2012. Phrase structures and dependencies for end-to-end coreference resolution. In *Proceedings of COLING 2012*.
- Susan E Brennan, Marilyn W Friedman, and Carl J Polard. 1987. A centering approach to pronouns. In *Proceedings of 25th ACL*.
- Praveen Dakwale and Himanshu Sharma. 2011. Anaphora resolution in indian languages using hybrid approaches. In *NLP tool contest, ICON 2011*.
- Praveen Dakwale, Himanshu Sharma, and Dipti M Sharma. 2012. Anaphora annotation in hindi dependency treebank. In *Proceedings of PACLIC-26*.
- Jerry Hobbs. 1986. Resolving pronoun references. In *Readings in natural language processing*. Morgan Kaufmann Publishers Inc.
- Itisree Jena, Riyaz Ahmad Bhat, and Sambhav Jain. 2013. Animacy annotation in hindi treebank. In *Proceedings of the LAWVII. ACL*.
- Igor Aleksandrovič Melčuk. 1988. *Dependency syntax: theory and practice*. State University of New York Press.
- Rashmi Prasad and Michael Strube. 2000. Discourse salience and pronoun resolution in hindi. *U. Penn Working Papers in Linguistics*.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27.
- Bhargav Uppalapu and Dipti Misra Sharma. 2009. Pronoun resolution for hindi. In *DAARC-7*.
- Ian H Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

# Structure Cognizant Pseudo Relevance Feedback

Arjun Atreya V, Yogesh Kakde, Pushpak Bhattacharyya, Ganesh Ramakrishnan

CSE Department, IIT Bombay, Mumbai

{arjun,pb, ganesh}@cse.iitb.ac.in, yrkakde@gmail.com

## Abstract

We propose a structure cognizant framework for pseudo relevance feedback (PRF). This has an application, for example, in selecting expansion terms for general search from subsets such as Wikipedia, wherein documents typically have a minimally fixed set of fields, *viz.*, *Title*, *Body*, *Infobox* and *Categories*. In existing approaches to PRF based expansion, weights of expansion terms do not depend on their field(s) of origin. This, we feel, is a weakness of current PRF approaches. We propose a per field EM formulation for finding the *importance* of the expansion terms, in line with traditional PRF. However, the final weight of an expansion term is found by weighting these *importance* based on whether the term belongs to the title, the body, the infobox or the category field(s). In our experiments with four languages, *viz.*, English, Spanish, Finnish and Hindi, we find that this structure-aware PRF yields a 2% to 30% improvement in performance (MAP) over the vanilla PRF. We conduct ablation tests to evaluate the importance of various fields. As expected, results from these tests emphasize the importance of fields in the order of title, body, categories and infobox.

## 1 Introduction

The ruling paradigm for Information retrieval (IR) (Manning et al., 2009) is *Pseudo Relevance feedback (PRF)*. In PRF, an assumption is made that the top retrieved documents are relevant to the query for picking expansion terms. Zhai and Lafferty (2001) show that using pseudo relevance feedback on monolingual retrieval improves the

overall result considerably over the retrieval without PRF. In case of retrieval for languages with little web content, Chinnakotla et al., (2010) show that taking help of another language to expand query helps in better performance.

The motivation for our work is as follows. Every document in the web collection has certain structure associated with it *viz.*, title, body, links, *etc.* Each of these fields has different level of importance in the document. For instance, document title broadly describes the whole document, whereas the body of the document contains the details. Content in these fields have different scales of contribution in uniquely representing that document in the collection. Hence it is important to consider the structure of a document while extracting expansion terms from it.

Structure based PRF, of course, draws on the basic theory of PRF as in Zhai and Lafferty (2001), which is based on expectation maximization (EM). We formulate a per field EM to get the weights of expansion terms and subsequently take their weighted sum in a spirit similar to mixture models.

## 2 Related Work

Approaches based on the use of external resources like wordnet for query expansion, though extensively studied, have been eventually dropped (Gong et al., 2005; Qiu and Frei, 1993). Several works have also used structure of documents for query expansion. These works propose the technique of first choosing relevant documents and finding expansion terms, therefrom, using co-occurrence, meta tags *etc.* Al-Shboul and Myaeng (2011) use categories of Wikipedia pages to cluster documents and retrieve the relevant cluster for query. This approach gives better recall at the cost of precision.

Anchor texts in Wikipedia pages pointing to a category same as the query category are picked

as expansion terms in Ganesh and Verma (2009). This work exploits the structure only in the form of anchor texts and category information.

Techniques to disambiguate query terms based on disambiguation pages of Wikipedia are proposed in (Xu et al., 2009; Lin et al., 2010). Once disambiguated, the page is considered for picking expansion terms. Other literatures that deal with PRF based IR are (Milne. et al., 2007; Lin and Wu, 2008; Lv and Zhai, 2010; Jiang, 2011).

### 3 Our System

We make use of Wikipedia as an external document collection for picking expansion terms. Reasons for this are: *a*) open source *b*) well-defined structure *c*) authenticity due to crowdsourcing and review, *d*) coverage across domains and languages *e*) ever growing. Four fields from the Wikipedia document are considered *viz.*, *title*, *body*, *categories* and *infobox*.

Our problem statement is:

*Given a query Q in a language L, retrieve relevant results from any document collection (WWW/dataset) in L using Wikipedia documents in L for generating expansion terms.*

The process of PRF based retrieval involves the following steps.

1. Retrieve ranked list of Wikipedia documents for a given query  $Q$ - *RetrievalModel* (Section 3.1)
2. Pick expansion terms from the top  $k$  retrieved documents- *ExpansionModel* (Section 3.2)
3. Obtain a modified query  $Q'$  by combining the expansion terms with the query terms- *AggregationModel* (Section 3.3)
4. Retrieve ranked list of documents for the modified query  $Q'$ - *RetrievalModel* (Section 3.1)

#### 3.1 Retrieval Model

Language model based retrieval is used in (Ponte and Croft, 1998) and (Croft, 2003). For every document  $D$ ,  $\theta_D$  is the probability distribution of terms. Similarly,  $\theta_Q$  is for the query  $Q$ . The "distance" between the query and a document,  $D_{KL}$  is calculated as equation 1.

$$D_{KL}(\theta_Q|\theta_D) = - \sum_w P(w|\theta_Q) \log P(w|\theta_D) \quad (1)$$

The more the relevance of  $D$ , the less is  $D_{KL}(\theta_Q|\theta_D)$ .

#### 3.2 Expansion Model

This model picks expansion terms that get combined with the query. Choosing expansion terms involves selecting a set of relevant documents and identifying terms that uniquely represent them. We use the retrieval model mentioned in section 3.1 to pick top  $k$  documents.

There exist many off-the-shelf expansion models to choose expansion terms from (Ganesh and Verma, 2009; Al-Shboul and Myaeng, 2011). None of these, however, exploit the structure of relevant documents. (Zhai and Lafferty, 2001) explain one of the state of art techniques to choose expansion terms using EM algorithm without considering the structure of a document. In Zhai and Lafferty (2001), a set of relevant documents  $R$  is retrieved and all terms in these documents are considered as observations. Since  $R$  is a subset of the document collection  $C$ , all terms in  $R$  also appear in  $C$ . Both  $R$  and  $C$  act as sources for generating terms.

Given a document, the content in each field of the document represents the document with different levels of importance. In our expansion model, we use Wikipedia as the source of expansion terms. Every Wikipedia document is composed of four fields *title*, *body*, *category* and *infobox*.

Expansion terms are picked independently from each field of the Wikipedia document. We run EM algorithm on each field as explained in Zhai and Lafferty (2001). We formulate an EM algorithm for picking expansion terms from *Title* field instead from a document as the whole. *Body*, *Categories* and *Infobox* fields follow the same formulation. The probability of all title terms in  $R$  ( $P_{R_{tk}}$ ) is maximized using EM algorithm. Similarly, body terms, category terms and infobox terms are also maximized.

The output of interest in an iterative EM algorithm is the set of expansion terms for every field. EM algorithm gives the weights of the expansion terms, indicating their importance. Weighted combination of these sets of expansion terms from different fields of the document leads to the final set of expansion terms. Empirically decided weights ( $\alpha$ 's) are used for combining expansion terms from different fields as shown in the equa-

	Dataset	Query set	No.of documents
English	FIRE 2010	76-125(50)	1,25,586
Spanish	ELRA-E0036	41-200(160)	4,54,045
Finnish	ELRA-E0036	91-250(160)	55,344
Hindi	FIRE 2010	76-125(50)	95,216

Table 1: Details of Experimental Setup; numbers in parenthesis indicate the number of queries

tion 2.  $\alpha_x$  indicates the importance given to the document field  $x$ .

$$P_{Rk} = \alpha_t \cdot P_{R_tk} + \alpha_b \cdot P_{R_bk} + \alpha_c \cdot P_{R_ck} + \alpha_i \cdot P_{R_ik} \quad (2)$$

where  $\alpha_t + \alpha_b + \alpha_c + \alpha_i = 1$

### 3.3 Aggregation Model

Once expansion terms are picked from Wikipedia documents, they are merged with initial query terms. Introducing expansion terms increases the possibility of topic drift for the intended information need. Hence, it is important to give more weight to query terms compared to expansion terms. The equation 3 indicates the aggregation of query  $Q$  with the expansion terms  $E$  to get the modified query  $Q'$  with  $\lambda$  as the weight given to the query over the expansion terms.

$$Q' = \lambda Q + (1 - \lambda)E \quad (3)$$

## 4 Experimental Setup

We conduct experiments to evaluate the effect of document structure on expansion terms, using ELRA-E0036<sup>1</sup>(part of CLEF) and FIRE 2010<sup>2</sup> datasets. Experiments are done in four languages, English, Spanish, Finnish and Hindi. Following are the set of experiments conducted:

*NORF- No relevance feedback*: This is the simplest form of retrieval without using any expansion.

*PRF- Pseudo relevance feedback without using the structure of a document*: This is traditional PRF. All terms in Wikipedia are considered to be equally important, and the naive expansion model of (Zhai and Lafferty, 2001) is used to find expansion terms.

*StructPRF- Pseudo relevance feedback using the structure of a document*: This is our proposed model. Structure of Wikipedia documents is used for finding expansion terms using the model described in section 3.2.

<sup>1</sup>[http://catalog.elra.info/product\\_info.php?products\\_id=1127](http://catalog.elra.info/product_info.php?products_id=1127)

<sup>2</sup><http://www.isical.ac.in/~fire/data.html>

	<i>NORF</i>	<i>PRF</i>	<i>StructPRF</i>
English	0.1758	0.2022 (+15%)	0.2189 (+24.5%)
Spanish	0.0433	0.1352 (+212%)	0.1778 (+310%)
Finnish	0.1532	0.2477 (+61.6%)	0.2517 (+64.3%)
Hindi	0.2321	0.2364 (+1.8%)	0.2529 (+9%)

Table 2: MAP scores; plus(+) indicates improvement over *NORF*

	<i>NORF</i>	<i>PRF</i>	<i>StructPRF</i>
English (2761)	1888	2080	2138
Spanish (2694)	391	1818	1919
Finnish (1377)	243	875	974
Hindi (915)	748	780	785

Table 3: Relevant documents retrieved; numbers in parenthesis indicate the actual relevant documents

Table 1 describes the experimental details. For every query, 1000 results are retrieved and used for evaluation. All languages use their respective Wikipedia content for picking expansion terms.

## 5 Results

MAP scores are shown in table 2. *StructPRF* has an overall improvement in MAP of 8% for English, 30% for Spanish, 2% for Finnish and 7% for Hindi over *PRF*. Figure 1 shows average precision values of all queries at different result positions for all languages. It is observed that there is a definite improvement in precision values for *StructPRF* over *PRF*. As we go down the list of retrievals ( $P@k$ , with  $k$  increasing), the improvement in *StructPRF* decreases but never gets below *PRF* and *NORF*.

Figure 2 depicts precision vs. recall curves for all languages. The results indicate that the *StructPRF* has a better precision for most recall val-

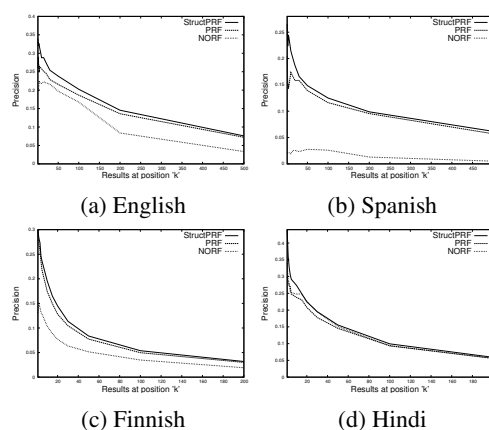


Figure 1:  $P@k$  Values

	English	Spanish	Finnish	Hindi
<i>NoTitle</i>	0.1953(-11%)	0.1179(-33%)	0.1914(-23%)	0.2086(-17%)
<i>NoBody</i>	0.2059(-6%)	0.1383(-22%)	0.2333(-8%)	0.2185(-13%)
<i>NoCategories</i>	0.2172(-0.7%)	0.1436(-19%)	0.2358(-7%)	0.2209(-12%)
<i>NoInfobox</i>	0.2178(-0.5%)	0.1467(-17%)	0.2449(-3%)	0.2234(-11%)

Table 4: MAP scores for ablation tests; minus(-) indicates percentage decrease from *StructPRF*

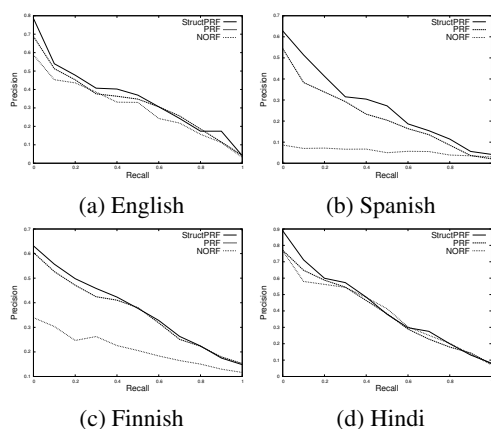


Figure 2: Precision-Recall Curve

ues. At 60% to 80% recall, precision of *PRF* is better than *StructPRF* in English. This indicates that most of the relevant documents are pushed higher up the order in the result set. For Spanish and Finnish, *StructPRF* consistently outperforms *PRF*. In Hindi, between 40% to 60% recall, *PRF* has a higher precision than *StructPRF*. This is again because of the relevant documents being pushed higher in the ranked list.

Analyzing query wise performances of *NORF*, *PRF* and *StructPRF* for all languages, we observed that *StructPRF* has best precision compared to other two for  $\approx 60\%$  of queries in all languages.

Table 3 shows that there is an improvement in the number of relevant documents retrieved by *StructPRF* compared to *PRF* for all languages. *StructPRF* has an improvement of 2.8%, 5%, 11% and 0.8% recall in English, Spanish, Finnish and Hindi respectively over *PRF*.

From these results it is evident that structure cognizant PRF benefits retrieval performance in terms of both precision and recall.

## 6 Ablation Tests

In ablation tests, we "disable" one field, that is, do not take expansion terms from a field, and get the MAP score. For instance, *NoTitle* has body, cat-

egories and infobox with equal weights (*i.e.*, 1/3) and weight of the title field as 0.

Table 4 lists the MAP scores for all cases of ablation. The name of each of these cases indicates the field "disabled". It is observed that the worst degradation in MAP occurs on disabling the *Title* field. This happens for all languages. The degradation decreases in the order of *Title*, *Body*, *Categories* and *Infobox*.

The above observation translates to setting values for  $\alpha_t$ ,  $\alpha_b$ ,  $\alpha_c$  and  $\alpha_i$  described in section 3.2 as  $\alpha_t > \alpha_b > \alpha_c > \alpha_i$  with  $\alpha_t + \alpha_b + \alpha_c + \alpha_i = 1$ . Hence the choice of  $\alpha$ 's for experimentation are 0.4, 0.3, 0.2 and 0.1 for  $\alpha_t$ ,  $\alpha_b$ ,  $\alpha_c$  and  $\alpha_i$  respectively.

The fields being important in the order of *Title*, *Body*, *Categories* and *Infobox* is quite intuitive. This is because the *Title* represents the content of the document with a few words. Hence, the *Title* field has a larger impact as compared to the *Body* field. Though *Categories* and *Infobox* have lesser words, like *Title*, they refer to a generic context of the query.

## 7 Conclusions and Future Direction

In this paper, we have explored the usage of document structure for PRF. We proposed an expansion model that considers each field of the document with different levels of importance in picking expansion terms. This structure cognizant PRF is compared with both traditional PRF and with no-feedback, for four languages, English, Spanish, Finnish and Hindi. Experimental results show that using structure helps in getting considerable improvement in both precision and recall over traditional PRF. Ablation tests reveal the relative importance of the fields, with "title" field proving more important than others.

In our work, we combine expansion terms obtained from every field of a document in a decoupled way, that is, through separate per field EMs. In future, we would like to explore tight coupling of document fields (EM over individual per-field EM).

## References

- Bashar Al-Shboul and Sung-Hyon Myaeng. 2011. Query phrase expansion using wikipedia in patent class search. In *AIRS*, pages 115–126.
- Manoj K. Chinnakotla, Karthik Raman, and Pushpak Bhattacharyya. 2010. Multilingual prf: english lends a helping hand. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '10*, pages 659–666, New York, NY, USA. ACM.
- W Bruce Croft. 2003. Language models for information retrieval. In *Proceedings of 19th international conference on data engineering*, pages 3–7.
- Surya Ganesh and Vasudeva Verma. 2009. Exploiting structure and content of wikipedia for query expansion in the context. In *International Conference RANLP*, pages 103–106.
- Zhiguo Gong, Chan Wa Cheang, and U Leong Hou. 2005. Web query expansion by wordnet. In *In DEXA*, pages 166–175.
- Xue Jiang. 2011. Query expansion based on a semantic graph model. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11*, pages 1315–1316, New York, NY, USA. ACM.
- Tien-Chien Lin and Shih-Hung Wu. 2008. Query expansion via wikipedia link. In *ITIA'08: The 2008 International Conference on Information Technology and Industrial Application*.
- Meng-Chun Lin, Ming-Xiang Li, Chih-Chuan Hsu, and Shih-Hung Wu. 2010. Query expansion from wikipedia and topic web crawler on clir. In *Proceedings of NTCIR-8 Workshop Meeting*, June 15-18.
- Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '10*, pages 579–586, New York, NY, USA. ACM.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2009. *An Introduction to information Retrieval*. Cambridge University Press, Cambridge, England.
- D Milne., Witten. I.H, and Nichols. D.M. 2007. A knowledge-based search engine powered by wikipedia. In *ACM Conference on Information and Knowledge Management*.
- Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of 21st annual international ACM SIGIR conference on research and development in information retrieval*, pages 275–281.
- Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '93*, pages 160–169, New York, NY, USA. ACM.
- Yang Xu, Gareth J.F. Jones, and Bin Wang. 2009. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '09*, pages 59–66, New York, NY, USA. ACM.
- Chengxiang Zhai and John Lafferty. 2001. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management, CIKM '01*, pages 403–410, New York, NY, USA. ACM.

# Cross-domain answer ranking using importance sampling

Anders Johannsen Anders Søgaard

University of Copenhagen

{ajohannsen, soegaard}@hum.ku.dk

## Abstract

We consider the problem of learning how to rank answers across domains in community question answering using stylistic features. Our main contribution is an importance sampling technique for selecting training data per answer thread. Our approach is evaluated across 30 community sites and shown to be significantly better than random sampling. We show that the most useful features in our model relate to answer length and overlap with question.

## 1 Introduction

Community Q&A (cQA) sites are rich sources of knowledge, offering information often not available elsewhere. While questions often attract the attention of experts, anyone can chip in, and as a result answer quality varies a lot (Fichman, 2011). cQA sites deal with this problem by engaging the users. If people like an answer or find it useful, they vote it up, and if it is wrong, unhelpful or spammy, it gets a down vote and is sometimes removed altogether. To a large degree the success of cQA can be attributed to this powerful content filtering mechanism. The voting induces a ranking of the answers, and that is the ranking we wish to reproduce in this paper.

We are interested in learning a ranking model based on textual or stylistic features only, extracted from the question and the answer candidate, because willfully ignoring information about user behavior and other social knowledge available in cQA sites makes our model applicable in a wider range of circumstances. Outside the world of cQA, automatic answer ranking might, for instance, be used to prioritize lists of answers found in FAQs or embedded in running text. In other words, we are interested in learning *a reranking model that is generally applicable to question answering systems*.

Part of what makes one answer preferable to another is how effective it is in communicating its advice. There may be plenty of answers that in some technical sense are correct and yet are not especially helpful. For instance, if the kind of advice we are looking for involves a procedure, an answer structured as “First ... Then ... Finally” would probably be of greater use to us than an answer with no discernible temporal structure. Our features capture aspects of the discourse surface structure of the answer. If the model is supposed to be generally applicable to question answering it also needs to exhibit *robust performance across domains*. Learning that mentions of specific Python modules correlate with answer quality in Stack Overflow does not help us answer questions in the cooking domain. We need to limit ourselves to features that transfer across domains. We further hypothesize a link between question type and answer structure (e.g. good answers to how-to questions look different from good answers to questions that ask for definitions), and test this experimentally by choosing training data for our ranker according to question similarity.

Our contribution is thus two-fold. We evaluate various stylistic feature groups on a novel problem, namely cross-domain community answer ranking, and introduce an importance sampling strategy that leads to significantly better results.

**Setup** Given a question and a list of answers the task is to predict a ranking of the answers matching the ranking induced by community voting. We approach this as a pairwise ranking problem, transforming the problem into a series of classification decisions of the form: does answer *a* rank ahead of *b*? We wish to train a model that maintains good performance across domains, and our evaluation reflects this goal. We use a leave-



one-out procedure where one by one each domain is used to evaluate the performance of a ranking model trained on the rest of the domains. Testing is thus always out-of-domain, and the setup promotes learning a generic model because the training set is composed of a variety of domains.

The rest of the paper is organized as follows. In the next section we introduce the cQA corpus. Section 3 describes several classes of motivated, domain-independent features. Our experiments with ranking and domain adaptation by similarity are described in Section 5, and the results are discussed in Section 6. Before the conclusion we review related work in Section 8.

## 2 The STACKQA corpus

We collected a corpus, the STACKQA corpus, consisting of questions paired with two or more answers from 30 individual cQA sites on different topics<sup>1</sup>. All sites are a part of the Stack Exchange network, sharing both the technical platform and a few very simple guidelines for how to ask a question. In the FAQ section of all sites, under the heading of "What kind of questions should I not ask here?", an identical message appears: "You should only ask *practical, answerable questions based on actual problems that you face*. Chatty, open-ended questions diminish the usefulness of our site and push other questions off the front page." It is, in other words, not a discussion club, and if a dubious question or answer enters the system, the community has various moderation tools at disposal. As a consequence, spam is almost non-existent on the sites.

## 3 Feature sets

Below we describe our six groups of features. Previous studies have shown that most of these features are correlated with answer quality, see (Jeon et al., 2006; Zhou et al., 2012; Harper et al., 2008; Su et al., 2010; Aji and Agichtein, 2010).

**Discourse** We use the discourse marker disambiguation classifier of Pitler and Nenkova (2009) to identify discourse uses. We have features which count the number of times each discourse marker appears.

**Length** This group has four features that measure the length of the answer in tokens and sen-

<sup>1</sup>We use the August 2012 dump from <http://www.clearbits.net/torrents/2076-aug-2012>

tences as well as the difference between the length of the question and the length of the answer. An additional two features track the vocabulary overlap between question and answer in number of lexical items, one including stop-words and one excluding these.

**Lexical diversity** An often used measure of lexical diversity is the type-token ratio, calculated as the vocabulary size divided by the number of tokens. We use a variation, the lemma-token ratio, which works on the non-inflected forms of the words.

**Level and style** For most readers understanding answers with long compound sentences and difficult words is a demanding task. We track difficulty of reading using the Flesch-Kincaid reading level measure and the closely related average sentence length and average token length. Three additional stylistic features capture the rate of inter-sentence punctuation, exclamation marks, and question marks. Finally, a feature gives the number of HTML formatting tokens.

**Pronouns** Scientific text almost never uses the pronoun "I", but other genres have different conventions. In cQA, where one person gives advice to another, "I" and "you" might feel quite natural. We capture personal pronoun use in six features, one for every combination of person and number (e.g. first person, singular).

**Word categories** These features build on groups of functionally related words. Examples of categories are transition words (213), which is a non-disambiguated superset of the discourse markers, phrases that introduce examples (49), comparisons (66), and contrast (6). Numbers in parenthesis indicates how many words there are in each category. For each category we count the number of token occurrences and the number of types.<sup>2</sup>

## 4 Importance sampling

The cQA sites contain abundant training data, but the sites are diverse and heterogeneous. We hypothesize that training our models on similar threads from different domains will improve our models considerably. We measure similarity with

<sup>2</sup>The word lists are distributed as a part of the LightSIDE essay assessment software package found at <http://lightsidelabs.com/>

respect to direct questions, disregarding any explanatory text. One complication is that the question text may have more than one sentence with a question mark after it—in fact, each thread contains 2.2 sentences ending with question marks, on average. To assess the similarity between two question threads  $Q$  and  $Q'$ , we take the maximum similarity between any of their question sentences:

$$\text{sim}(Q, Q') = \max_{q \in Q, q' \in Q'} \text{sim}(q, q')$$

The similarity function used is a standard information retrieval TF\*IDF-weighted bag-of-words model. Table 1 shows an example of the similar questions found by this method.

Since importance sampling requires a separately trained classifier for each question thread, we evaluate on a small set of 500 question threads per domain.

## 5 Experiments

For each site we sample up to 5000 question threads that contain between 2 and 8 answers. When more than one answer have the same number of votes, making it impossible to rank the answers unambiguously, one of the tied answers is kept at random. The number of threads used for training is varied from 50 to 5000 to obtain learning curves. We compare importance sampling against random sampling. Because this procedure is random, we repeat it three times and report an average performance figure.

The baseline for evaluating our feature model is a TF\*IDF weighted bag-of-words model with each answer normalized to unit length.

We rank the answers by applying the pairwise transformation (Herbrich et al., 1999) and learn a classifier for the binary relation  $\prec$  (“ranks ahead of”). Training data consists of comparisons between pairs of answers in the same thread.

We report  $F_1$  score for the binary discrimination task and Kendall’s  $\tau$  for the ranking. In Kendall’s  $\tau$  1.0 means perfect fidelity to the reference ordering, -1.0 is a perfect ordering in reverse, and .0 corresponds to a random ordering.

## 6 Results

Table 3 shows that importance sampling leads to significantly better results.

The ablation results in Table 2 show that the largest negative impact comes from removing the

### Question

---

**How do you clean a cast iron skillet?** (Cooking)  
 How do you clear a custom destination? (Gaming)  
 How do you restore a particular table in MySQL? (DB)  
 How Do You Determine Your Hourly Rate? (Programmers)  
 Do you know how to do that? (Unix)  
 How do I do this? (Gaming)  
 How do you select the Fourth kill streak? (Gaming)  
 How do you deal with unusually long labels? (Ux)  
 How do I delete a tumblr blog? (Web apps)  
 How do you use your iPod shuffle or nano? (Apple)  
 So, how do you explain spinning tops to a nine year old? (Physics)

Table 1: The 10 questions most similar to the question in bold, not counting questions from the same domain.

	F1	$\tau$
Full model	.593	.210
- lexical diversity	.592	.209
- discourse	.605	.235
- length	.555	.136
- level and style	.592	.211
- pronouns	.593	.210
- word categories	.600	.226

Table 2: Feature ablation study on the importance weighted system (System+Sim). The results are for a training set of 500 threads.

length-related features. Leaving them out, the performance drops to .136 (from .210) in the ranking fidelity measure.

## 7 Discussion

The fact that no feature group independently contributes to the classification performance, apart from the length related features, is interesting, but note that even with the length related features removed, the system is still significantly better than the bag-of-words baseline.

The relatively low performance raises two questions, discussed below. How much trust should we put into the user rankings, which are the gold standard in the experiments? And what is the maximum performance we can expect?

There is no guarantee that people who submit votes are experts. For this reason, Fichman (2011) dismiss the “best answer” feature of cQA, adding that askers often select the best answer guided by social or emotional reasoning, rather than by facts. In a case study on Stack Overflow (part of the StackExchange network), Anderson et al. (2012)

Thread count	Kendall's $\tau$			F1		
	Baseline	System	System+Sim	Baseline	System	System+Sim
50	.070	.075	.099	.355	.522	.536
100	.107	.084	.129	.381	.528	.551
250	.121	.095	.166	.518	.533	.571
500	.135	.124	.199	.529	.549	.588
1000	.146	.158	.229	.557	.566	.603
5000	.161	.215	<b>.253</b>	.578	.595	<b>.615</b>

Table 3: Ranking performance. Baseline is a bag-of-words model, and System uses the full feature set described in the paper. System+Sim uses the same feature model as System but with importance sampling. Results are an average over domains, and all differences between System+Sim and System are significant at  $p < .01$  using the Wilcoxon ranksum test.

find that voting activity on a question is influenced by a number of factors presumably not connected to answer quality, such as the time before the first answer arrives, and the total number of answers.

With respect to the maximum attainable performance, an important consideration is that an answer is judged on other factors than how well it is written. When seeking a solution to a practical problem, the best answer is the one that solves it, no matter how persuasive the other answers are. This holds particularly true for cQA sites that advise people only to ask questions related to actual, solvable problems. The textual model is strong mainly if we have multiple alternative answers, which are indistinguishable with respect to facts, but differ in how their explanations are structured.

## 8 Related work

Moschitti and Quarteroni (2011) consider the problem of reranking answers in question-answering systems. They use kernelized SVMs, noting that the kernel function between (question, answer) pairs can be decomposed into a kernel between questions and a kernel between answers:  $K(\langle q, a \rangle, \langle q', a' \rangle) = K(q, q') \oplus K(a, a')$ . They share the intuition behind our approach, that pairs with more similar questions should have higher weight, but we sample data points instead of weighting them and use different similarity functions. Choi et al. (2012) establish a typology of questions in social media, identifying four different varieties: information-seeking, advice-seeking, opinion-seeking, and non-information seeking. For our purposes their categories are probably too broad to be useful, and they require manual annotation.

Agichtein et al. (2008) identify high quality answers in the Yahoo! Answers data set. In addition to a wide range of social features, they have three groups of textual features: punctuation and typos, syntactical and semantic complexity, and grammaticality.

Shah and Pomerantz (2010) evaluate answer quality on Yahoo! Answers data. They solicit quality judgements from Amazon Mechanical Turk workers who are asked to rate answers by 13 criteria, such as readability, relevancy, politeness and brevity. The highest classification accuracy is achieved using a combination of social and text length features.

Lai and Kao (2012) address the problem of matching questions with experts who are likely to be able to provide an answer. Their algorithm is tested on data from Stack Overflow.

He and Alani (2012) investigate best answer prediction using StackExchange's Serverfault and cooking communities as well as a third site outside the network.

## 9 Conclusion

In this paper we report on experiments in cross-domain answer ranking. For this task we introduced a new corpus, a feature representation and an importance sampling strategy. While the questions and answers come from a cQA setting, models learned from this corpus should be more widely applicable.

## Acknowledgements

We wish to thank the ESICT project for partly funding this work. The ESICT project is supported by the Danish Council for Strategic Research.

## References

- Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining*, pages 183–194. ACM.
- Ablimit Aji and Eugene Agichtein. 2010. The nays have it: exploring effects of sentiment in collaborative knowledge sharing. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*.
- Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Erik Choi, Vanessa Kitzie, and Chirag Shah. 2012. Developing a typology of online Q&A models and recommending the right model for each question type. In *HCIR 2012*, number 3.
- Pnina Fichman. 2011. A comparative assessment of answer quality on four question answering sites. *Journal of Information Science*, 37(5):476–486.
- F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. 2008. Predictors of answer quality in online Q&A sites. In *CHI '08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
- Yulan He and Harith Alani. 2012. Automatic Identification of Best Answers. In *9th Extended Semantic Web Conference 2012*, pages 514–529.
- Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 1999. Support vector learning for ordinal regression. In *9th International Conference on Artificial Neural Networks: ICANN '99*, volume 1999, pages 97–102. IEE.
- Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.
- Liang-Cheng Lai and Hung-Yu Kao. 2012. Question Routing by Modeling User Expertise and Activity in cQA services. In *The 26th Annual Conference of the Japanese Society for Artificial Intelligence*.
- Alessandro Moschitti and Silvia Quarteroni. 2011. Linguistic kernels for answer re-ranking in question answering systems. *Information Processing & Management*, 47(6):825–842, November.
- Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 13–16, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Chirag Shah and Jefferey Pomerantz. 2010. Evaluating and predicting answer quality in community QA. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 411–418, New York, NY, USA. ACM.
- Qi Su, Chu-Ren Huang, and Helen Kai-yun Chen. 2010. Evidentiality for text trustworthiness detection. In *NLPLING '10 Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground*.
- Zhi-Min Zhou, Man Lan, Zhen-Yu Niu, and Yue Lu. 2012. Exploiting user profile information for answer ranking in cQA. *WWW '12 Companion Proceedings of the 21st international conference companion on World Wide Web*.

# Morphological Analysis of Tunisian Dialect

**Inès Zribi**  
ANLP Research group,  
MIRACL Lab.,  
University of Sfax, Tunisia  
ineszribi@gmail.com

**Mariem Ellouze Khemakhem**  
ANLP Research group,  
MIRACL Lab.,  
University of Sfax, Tunisia  
mariem.ellouze@planet.tn

**Lamia Hadrich Belguith**  
ANLP Research group,  
MIRACL Lab.,  
University of Sfax, Tunisia  
l.belguith@fsegs.rnu.tn

## Abstract

In this paper, we address the problem of the morphological analysis of an Arabic dialect. We propose a method to adapt an Arabic morphological analyzer for the Tunisian dialect (TD). In order to do that, we create a lexicon for the TD. The creation of the lexicon is done in two steps. The first step consists in adapting a Modern Standard Arabic (MSA) lexicon. We adapted a list of MSA derivation patterns to TD. The second step consists in improving the resulting lists of patterns and roots by using TD specific roots and patterns. The proposed method has been tested and has achieved an F-measure performance of 88%.

## 1 Introduction

The Arabic Dialect (*AD*) is a collection of spoken varieties of Arabic. It is used in everyday communication. So, it is so important to consider it in the new technologies like dialogue systems, telephone applications, etc. (Zribi et al., 2013). The majority of these applications need a morphological analysis to segment words and to exploit their morphological features.

Many important works have focused on the morphological analysis of the Arabic language, mainly on Modern Standard Arabic (*MSA*). *AD* has not received much attention due to the lack of dialectal tools and resources (Duh and Kirchhoff, 2005). However, there are differences between *MSA* and *AD*, they are considered as two related languages.

Therefore, we suggest in this paper to exploit and adapt an *MSA* morphological analyzer (*MA*) to Tunisian Dialect (*TD*). The adaptation is done in two steps. The first step is to adapt an *MSA* lexicon to *TD* and to improve the resulting lexicon with *TD* specific roots and derivation patterns<sup>1</sup>. The second step is to integrate the

resulting lexicon into the *MSA* morphological analyzer.

The paper has 5 main sections. Section 2 presents a lexical study of the *TD*. We present in section 3 an overview of previous works. We describe in section 4 our method for adapting *MSA* resources to *TD*. In section 5, we give the results of the system evaluation, and finally, we discuss some analysis errors.

## 2 TD lexical study

*TD* is characterized by a phonology, a morphology, a syntax and a lexicon which have differences and similarities compared to *MSA* and even to other Arabic dialects (Zribi et al., 2013). There are many regional varieties. In this paper, we focus on the standard *TD* (the dialect used in the media that is the most understood by all Tunisians).

### 2.1 STAC corpus presentation

In order to develop and test our method, we created the *STAC* corpus by recording and manually transcribing some radio and TV broadcasts. *STAC* corpus consists of 3 hours and 20 minutes of speech. The corpus relates to various fields: politics, health, social issues, religious issues and others. *STAC* corpus is composed of about 27,144 words. We used  $\frac{3}{4}$  of the corpus for the training of our method. This portion of the corpus contains 443 distinct nouns and 235 distinct verbs. We used the rest of the corpus to test the performance of our system (see section 5). We used *OTTA* conventions (Zribi et al., 2013) while transcribing and annotating our *STAC* corpus. It is to be noted that we respect in this paper the *OTTA* conventions (Zribi et al., 2013) when writing examples of words in *TD*.

<sup>1</sup> The Arabic derivation system consists to use a set of patterns and roots to generate words. To generate the word “يكتب”, *yaktibu*, *he write*, we replace the *r<sub>i</sub>* letters in the

following pattern “*yar<sub>1</sub>r<sub>2</sub>ir<sub>3</sub>u*” with the letters of the root “*ktb*” by respecting the order of letters. (a,i,u) represent the Arabic short vowels. The Arabic orthographic transliteration used in this paper is presented in (Habash et al., 2007).

## 2.2 Classification of the TD lexicon

The linguistic study of the words composing our TD corpus shows that its lexicon can be classified into four classes. *The first class (C1)* includes words that are derived from MSA roots via the application of the derivation patterns of MSA. These patterns are generally modified compared to those of MSA. They witness small changes, mainly, in vowels and in some letters forming these patterns (some letters are added, deleted or modified). For example, the derivation patterns (يُفَعِّلُ,  $yir_1r_2ar_3$ ) and (فَعَّيَاةُ,  $r_1r_2aAyap$ ) are the result of some changes of the pattern (يُفَعِّلُ,  $yar_1r_2ar_3u$ ) and the pattern (فَعَّيَاةُ,  $r_1ir_2Ar_3ap$ ). This class presents 85.13% of our STAC corpus. *The second class (C2)* includes words that are derived from TD specific roots via the application of patterns following the MSA derivation patterns (the patterns used in *(C1)*). For example, the verb (يَنْقُزُ,  $ynagiz$ , “he jumps”) is derived from the Tunisian root (نَقَزَ,  $ngz$ ) and the pattern (يُنْفَعِّلُ,  $yr_1ar_2ir_3$ ). This class presents 8% of our STAC corpus. *The third class (C3)* includes words that are derived from the MSA roots with the application of TD specific patterns. These patterns do not match with MSA patterns. For example, the word (فَهْوَاجِي,  $qahwaAjiv$ , “a waiter”) is derived from the MSA root (فَهْو,  $qhw$ ) and the derivation pattern (فَعَّلَاجِي,  $r_1ar_2r_3aAjiv$ ). This class presents 4.26% of our STAC corpus. *The fourth class (C4)* contains words which are derived from foreign languages specifically French. For example, the word (يُدَوِّشُ,  $ydawis$ ) is derived from the French sentence (il prend une douche, “he is having a shower”). This class presents 2.62% of our STAC corpus.

From this study, we deduce that to create a TD lexicon, we should determine the list of TD patterns and apply them to the list of MSA roots, or determine the list of TD roots, and apply the patterns of the MSA to generate a TD lexicon.

## 3 Related works

Arabic dialects can be considered as under-resourced languages because of the absence of tools and resources. Therefore, we will study some works dealing with the automatic processing of under-resourced languages. Among these works, we cite the works of Borin (2002), Das and Petrov (2011), Lindström and Müürisep (2009), Shalónova and Golénia (2010), etc. Some of these works ((Rambow et al, 2005), (Lindström and Müürisep, 2009), (Das and Petrov, 2011), etc.) are based on resources and

tools of cognate languages that are adapted for the processing the under-resourced language. Other works ((Yang et al., 2007), (Shalónova and Golénia, 2010), etc.) are based on a small amount of data for the analysis of under-resourced languages. Hana (2008) adopted this approach to propose a method for the morphological analysis of Czech language. He used a small list of words accompanied by information about their lemma and tags to develop a Guesser. The role of the Guesser is to deduce from a corpus the lemma-stem-paradigm candidates for each unknown word. These candidate paradigms are, then, validated and added to a lexicon. Hana (2008) utilized the resulting lexicon for developing a Czech MA. Some works have tackled the task of the morphological analysis of AD. The general idea of these works is to adapt existing tools designed for MSA. Among these works, we cite the work of Salloum and Habash (2011) and Almeman and Lee (2012) who added a list of dialectal affixes to two MSA MA (*BAMA* (Buckwalter, 2004) and *Al-Khalil* (Boudlal et al, 2011)). Habash et al. (2012) transformed an Egyptian dialect lexicon into a tabular form that is compatible the MA SAMA (Graff et al., 2009). Habash and Rambow (2006) developed *MAGEAD*, a MA for the Arabic language and its dialects.

We propose in this work to adapt a MSA MA. We propose first to adapt an MSA lexicon to TD and to improve the resulting lexicon with TD specific roots and patterns. Then, we integrate the resulting lexicon into the MSA MA.

## 4 Adapting MSA resources to TD

Our goal in this paper is to develop a TD morphological analyzer taking advantage of the existing resources of the Arabic language. Like previous works on AD morphological analyzers, we propose to adapt an existing MA analyzer and to create the necessary resources for its adaptation. We do not limit to add dialectal affixes, but we propose to incorporate a lexicon to a MA for MSA. Given that we don't have such a lexicon, we exploit the points of similarity between TD and MSA for its creation. First, we start from MSA lexicon to generate a small lexicon for the TD. We use this list for building a TD lexicon. The process of creation of TD lexicon is similar to the ending-based Guesser module of Hana (2008) that suggests a lemma-stem-paradigm candidate for each word in the corpus. Our method for building our TD lexicon is composed of

two main steps: the transformation of MSA patterns into TD patterns, and the extraction of TD specific roots and patterns. We detail in this section the different steps of our method. Then, we present the list of TD function words, affixes and clitics.

**Transformation of MSA patterns into TD patterns:** The first step of our method consists in determining from a set of MSA patterns the corresponding patterns in TD. Firstly, we classify the roots of the lexicon. Indeed, the Arabic roots can be classified according to several criteria: the number of root letters, the presence and the number of defective letters, etc. We adopt in our work the classification based on the presence and the number of defective letters. The study of TD morphology done by Ouerhani (2009) shows that the verbs belonging to the Mahmoudz class (which includes the roots containing the letter ء) share the same patterns and features and follow the same rules when they are transformed in TD. This deduction is also applicable to other root classes. For example, the verbs (بدأ, *badā>a*, “he started”) and (ملا, *malā>a*, “he filled”) in MSA that have respectively the roots (بدء, *bd'*) and (ملء, *ml'*) are transformed into (بدأ, *bdA*) and (ملا, *mlA*) in TD. These verbs follow the same derivation pattern in MSA (فَعَلَ, *r<sub>1</sub>ar<sub>2</sub>ar<sub>3</sub>a*) as in TD (فَعَا, *r<sub>1</sub>r<sub>2</sub>aA*) keeping the same morphological features. For each class of roots and for each MSA pattern, we determine the corresponding TD derivation pattern(s) and, we update their lists of features. In the case of verbs, we determine for each person and for each aspect, the different patterns in TD. For example, the MSA pattern (فَعَلَ, *r<sub>1</sub>ar<sub>2</sub>ir<sub>3</sub>a*) belonging to the Defective class (which includes the roots ending with defective letters) is transformed into (فَعَى, *r<sub>1</sub>r<sub>2</sub>aY*) in TD. In the case of nouns, we determine for each type (noun, adjective, etc.) and for each gender, the different patterns in TD. For example, for the Mahmoudz class, the derivation pattern (فَاعِلَةٌ, *r<sub>1</sub>aAr<sub>2</sub>ir<sub>3</sub>ap*) in MSA is transformed to (فَائِلَةٌ, *r<sub>1</sub>Ayr<sub>3</sub>ap*) in TD. The result of this step is a TD lexicon composed of 6,092 patterns (nominal and verbal) and 6,030 roots. Six hundred and fifty patterns were kept from the MSA lexicon during this step. This lexicon covers the first class (C1).

**Root and pattern extraction:** After converting MSA patterns to TD, the next step is intended to enhance the coverage of TD lexicon. This step is composed of two phases. The first phase consists in extracting TD specific roots. The aim of this step is to cover the second class (C2). We try to extract roots from a training

corpus which contains specific TD words such as the verb (نَجَزَ, *nagiz*, “he jumped”) and the noun (كَرْهَبَةٌ, *karhbap*, “a car”). To perform these tasks, we proceed as follows: We analyze all the words of the corpus using the lexicon generated in the first step. If there is no analysis, we try to extract roots for these words corresponding to patterns derived from the first step. The extracted roots are saved in a temporal list. If the frequency of a root is greater than two, we add this root to the lexicon. For example, using the verbal patterns (يَفْعَلُ, *yar<sub>1</sub>r<sub>2</sub>ir<sub>3</sub>*) and (فَعَّلَ, *r<sub>1</sub>ar<sub>2</sub>r<sub>3</sub>ir<sub>4</sub>*), we can extract respectively the root (نَجَزَ, *ngz*) and the root (يَنْجِزُ, *yngz*) for the unrecognized word (يَنْجِزُ, *ynagiz*, “he jumps”). Similarly, using the nominal patterns (تَفْعِيلَةٌ, *tar<sub>1</sub>r<sub>2</sub>ir<sub>3</sub>ap*) and (تَفَعَّلَةٌ, *tr<sub>1</sub>ar<sub>2</sub>r<sub>3</sub>ir<sub>4</sub>ap*), we can extract respectively the root (نَجَزَ, *ngz*) and the root (نَجِيزُ, *ngyz*) for the unknown word (تَنْجِيزَةٌ, *tangyzap*, “a jump”). The frequency of the root (نَجَزَ) is equal to 2. So, we consider that the root of the words (نَجِيزَةٌ and يَنْجِزُ) is (نَجَزَ). In the second phase, we adopt the same idea as in the first phase. It consists to extract TD specific patterns. We aim in this step to cover the third class (C3). We use in this phase the list of roots derived from the first step. The difference between the root extraction step and this step is the validation of generated patterns by an expert. The expert determines the morphological features corresponding to the patterns.

**Function words and affixes:** Based on our training corpus and the MSA lexicon, we determined the list of TD clitics and the list of function words. From the MSA lexicon, we determined the possible translations for each function word. We noted that some MSA function words are transformed into TD function word(s) and/or clitics but some others cannot be translated to TD. For example, the future particle (سوف, *swf*, “I will”) can be translated to (باش, *bA\$*, “I will”). However, the Arabic preposition (من, *mn*, “from”) keeps the same form in TD but in some cases it is transformed to a proclitic (م, *m*, “from”). Similarly, we determine from the MSA lexicon the possible translations for each affix and clitic. Some MSA clitics are transformed into TD function word(s) and/or clitics and some others don't have an equivalent in TD. For example, the future prefix (س, *s*, “I will”) is transformed in TD to (باش, *bA\$*, “I will”). However, the interrogation prefix “أ” is transformed to a suffix (شي, *\$y*, “what”). We note also the definition of many other affixes and clitics: such as the new form (و, *w*, “him”) of the enclitic (ه, *h*,

“him”). We obtained 289 function words, and 66 affixes and clitics for the TD.

## 5 Implementation and evaluation

We have chosen the MSA MA *Al-Khalil* (Boudlal et al, 2011) to adapt it for analyzing TD. We selected *Al-Khalil* because it had been elated the best MA among ten morphological analyzers in a competition held in ALESCO in 2010. We also used its lexicon for generating the TD lexicon. It is composed of 7,503 roots and 3,681 unvoveled patterns. To enable *Al-Khalil* to analyze TD, we integrated the TD lexicon in its morphological analysis process. Moreover, we added new rules in the process of word tokenization (e.g. rules for segmenting the new enclitic “ج”). We have corrected, also, some gaps in *Al-Khalil* (e.g. no difference between affixes and clitics in the segmentation process).

### 5.1 Results and discussion

To test the performance of our system, we used our training corpus STAC (see section 2.1). To our knowledge, there is no existing TD MA to compare with, we, therefore, used the MSA version of *Al-Khalil* as a baseline to compare the performance of our TD MA. The system’s performance is evaluated in two ways. Firstly, the system is tested according to the number of words recognized by the analyzer. We calculate the number of words for which the analyzer attributes at least one correct analysis. The objective of this evaluation is to measure the analyzer’s coverage of the different classes of the TD lexicon. To measure the correctness of the results given by our TD MA, we tried another evaluation. The system’s performance was evaluated with reference to the generated analysis. We calculated the number of correct analyses given for each word. An analysis is considered correct if all of its features (part-of-speech, mood, gender, number, root, pattern, etc.) are fully correct.

In the evaluation process, we calculated the performance of the system using the TD lexicon generated in the first step of our method. Then, we evaluated the system using the lexicon resulting from the second step. Table 1 shows the results of the evaluation.

	Baseline		Step 1		Step 2	
	Eval1	Eval2	Eval1	Eval2	Eval1	Eval2
Recall	54%	70%	78%	86%	79%	89%
Precision	70%	65%	52%	60%	77%	80%
F-measure	54%	63%	67%	77%	78%	88%

Table 1 : Evaluation results

The two evaluations processes show that the second step of creation of the lexicon has improved the result of the TD MA. First, we used the MSA version of *Al-Khalil* (Boudlal et al, 2011). We obtained an F-measure equal to 63%. This result justifies the importance of the shared part between MSA and TD. Then, the evaluation of the first step of our method shows an improvement in the results. Indeed, the system can cover 86% of the words of the test corpus. We obtained an improvement of about 14% in F-measure metric. Finally, the evaluation of the system by using the lexicon resulting from the second step shows also an improvement of results. We got an overall F-measure equal to 88%. The results show clearly that the extraction root and pattern module has an improvement effect (about 10%). The failure in the analysis of some words can be explained by the lack of some patterns and/or roots in the training corpus. In addition, the wrong extraction of roots presents another cause of analysis failure. Certainly, the errors generated by the step of extraction of roots were caused by the use of the same patterns for different root classes. As a consequence, the system proposed different roots for the same conjugated verb. Some incorrect roots were added to the lexicon. So these wrong roots increase the number of incorrect analyses of some words. For example, the extraction of roots module has extracted the root (ينقر, *ynagiz*, “he jumps”) from the verbs (ينقر, *ynagiz*, “he jumps”) and (ينقروا, *ynagzuwA*, “they jump”). This root is automatically added to the TD lexicon. We note that the root (ينقر, *yngz*) is wrong. Other cases of failure were caused by the foreign origin of certain words (words derived from foreign languages such as French).

## 6 Conclusion

In this paper, we have proposed an original method to create a lexicon for TD. This method is based on two steps: the first step converts MSA patterns to TD ones; the second step extracts roots and patterns. The resulted lexicon was integrated in the *Al-Khalil* MA. This system has shown encouraging results (F-measure = 88.86%). As for our perspectives, we intend to extend the TD lexicon by covering words derived from foreign languages. Then, we plan to develop a module allowing the disambiguation of the output of the system by applying machine learning techniques.



## References

- Almeman, K., and Lee, M. 2012. *Towards Developing a Multi-Dialect Morphological Analyser for Arabic*. 4th International Conference on Arabic Language Processing, May 2–3, 2012, Rabat, Morocco (pp. 19–25).
- Borin, L. 2002. *Alignment and tagging*. Selected papers from a symposium on parallel and comparable corpora at Uppsala University (pp. 157–167). Amsterdam, Rodopi.
- Boudlal, A., Lakhouaja, A., Azzeddine, M., and Abdelouafi M. 2011. *Alkhalil Morpho Sys1: A Morphosyntactic analysis System for Arabic texts*. Proceedings of ACIT'2010, Riyadh, Saudi Arabia.
- Buckwalter, T. 2004. *Buckwalter Arabic morphological analyzer version 2.0*. LDC catalog number LDC2004L02, ISBN 1-58563-324-0.
- Das, D., and Petrov, S. 2011. *Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections*. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, (pp. 600–609). Portland, Oregon.
- Duh, K. and Kirchoff, K. 2005. *POS Tagging of Dialectal Arabic: A Minimally Supervised Approach*. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Ann Arbor, June 2005.
- Habash, N. Soudi, A. and Buckwalter, T. 2007. *On Arabic Transliteration*. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Soudi, Abdelhadi; van den Bosch, Antal; Neumann, Günter (Eds.), 2007. ISBN: 978-1-4020-6045-8.
- Habash, N., Eskander, R., and Hawwari, A. 2012. *A Morphological Analyzer for Egyptian Arabic*. Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON2012). (pp. 1–9). Montréal, Canada.
- Habash, N., Rambow, O., and Kiraz, G. 2006. *Morphological Analysis and Generation for Arabic Dialects*. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Ann Arbor (pp. 17–24).
- Hana, J. 2008. *Knowledge and Labor-Light morphological analysis*. OSUWPL, 58, 52–84.
- Ouerhani, B. 2009. *Interférence entre le dialectal et le littéral en Tunisie : Le cas de la morphologie verbale*. Synergies Tunisie, 1, 75–84.
- Rambow, O., Chiang, D., Diab, M., Habash, N., Hwa, R., Sima'an, K., Lacey, V., Levy, R., Nichols, C. and Shareef, S. 2005. *Parsing Arabic dialects*. Final Report, 2005 JHU Summer Workshop.
- Salloum, W., and Habash, N. 2011. *Dialectal to Standard Arabic Paraphrasing to Improve Arabic-English Statistical Machine Translation*. Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK (pp. 10–21).
- Shalonova, K., and Golénia, B. 2010. *Weakly Supervised Morphology Learning for Agglutinating Languages Using Small Training Sets*. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010) (pp. 976–983). Beijing.
- Yang, M., Zheng, J., Kathol, A. 2007. *A Semi-Supervised Learning Approach for Morpheme Segmentation for an Arabic Dialect*. Proceedings of Interspeech 2007.
- Zribi, I., Graja, M., Khmekhem, M. E., Jaoua, M., and Belguith, L. H. 2013. *Orthographic Transcription for Spoken Tunisian Arabic*. CICLing 2013, Part I, LNCS 7816 (pp. 153–163).

# Disambiguating explicit discourse connectives without oracles

Anders Johannsen Anders Søgaard  
University of Copenhagen  
{ajohannsen, soegaard}@hum.ku.dk

## Abstract

Deciding whether a word serves a discourse function in context is a prerequisite for discourse processing, and the performance of this subtask bounds performance on subsequent tasks. Pitler and Nenkova (2009) report 96.29% accuracy ( $F_1$  94.19%) relying on features extracted from gold-standard parse trees. This figure is an average over several connectives, some of which are extremely hard to classify. More importantly, performance drops considerably in the absence of an oracle providing gold-standard features. We show that a very simple model using only lexical and predicted part-of-speech features actually performs slightly better than Pitler and Nenkova (2009) and not significantly different from a state-of-the-art model, which combines lexical, part-of-speech, and parse features.

## 1 Introduction

Discourse relations structure text by linking segments together in functional relationships. For instance, someone might say "Saber-toothed tigers are harmless *because* they're extinct", making the second part of the sentence serve as an explanation for the first part. In the example the discourse connective *because* functions as a lexical anchor for the discourse relation. Whenever an anchor is present we say that the discourse connective is *explicit*.

Complicating the matter, phrases used as discourse connectives sometimes appear in a non-discourse function. For instance, "and" may be either a simple conjunction, as in "sugar and salt", or a discourse relation suggesting a temporal relationship between events, for instance "he struck the match and went away". The Penn Discourse

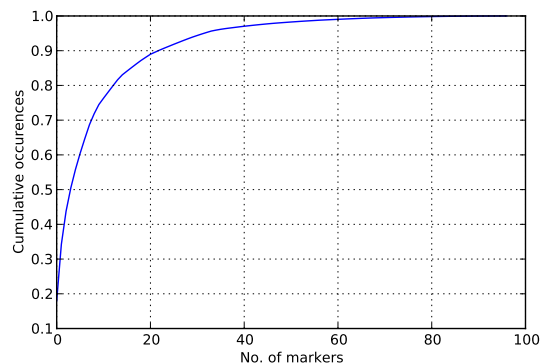


Figure 1: A picture of the problem. 10% of connectives account for roughly 75% of occurrences

Treebank (PDTB) (Prasad et al., 2008) distinguish 100 types of explicit connectives—a subset of these are listed in Table 2. The type of relationship is selected from a hierarchical structure where the four top-level categories are Comparison, Contingency, Temporal, and Expansion.

Discourse relations are important for many applications and, since the PDTB was released, much effort has gone into developing tools for recreating the annotations of the resource automatically. Recently two ambitious end-to-end parsers have appeared which transform plain text to full PDTB-style annotations (Lin et al., 2010; Ghosh, 2012). Both systems share a pipelined architecture in which the output of one component becomes the input to the next. A crucial first step in their processing is correctly identifying explicit discourse connectives; when unsuccessful subsequent steps fail.

An accuracy in the high nineties seems to suggest that the problem is almost solved. For the task of discourse connective disambiguation this unfortunately does not hold true, because, as we argue here, the task benefits from being seen and evaluated as a number of smaller tasks, one for each

connective type. Figure 1 shows why: the distribution of connectives follows a power law such that the majority of occurrences comes from relatively few but highly frequent connective types. If we do not take into account the uneven sizes of the categories, our performance figure ends up saying very little about how well we are doing on most of the connectives, because it is being dominated by the performance on a few high-frequency items.

In this paper we look in more detail on the evaluation of the discourse connective disambiguation task, in particular how two commonly used feature models perform on individual discourse connectives. The models are Pitler and Nenkova (2009) (P&N), and its extension by Lin et al. (2010) (Lin). Motivated by our findings we advocate the use of macro-averaging as a necessary supplement to micro-averaging. Additionally, we perform our experiments in a more realistic setting where access to oracle gold-standard annotations is not assumed. The observed performance drop from oracle to predicted parses leads us to propose a new model, which approximates the syntactical information of the parse trees with part-of-speech tags. Although these features are less powerful in theory, the model has comparable macro-average performance in realistic evaluation.

The rest of the paper is structured as follows. In the next section we give reasons why low-frequency connectives should not be overlooked. Section 3 describes our experiments, and Section 4 reports on the results. The discussion is in Section 5, followed by a review of related work in Section 6. Section 7 concludes the paper.

## 2 The importance of the long tail

Are there any compelling reasons to pay attention to the lower-frequency connectives when high-frequency connectives overwhelmingly dominate? As noted in the caption to Figure 1, the top 10 account for above 75% of the occurrences and top 20 for above 90%. So why should we care?

It turns out that the low-frequency connectives are quite evenly distributed among texts. In the Wall Street Journal part of the Penn Treebank, 70% of articles that contain explicit markers contain at least one marker not in the top 10. Not counting very short texts (having only two or fewer explicit connectives of any type), the number rises to 87%. While low performance on less frequent connectives does not hurt a token-level

macro-average much, it still means that you are likely to introduce errors in something like 70% of all WSJ articles. These errors percolate leading to erroneous text-level discourse processing.

In Webber and Joshi (2012) the prime example of a discourse application is automatic text simplification. Here, ignoring the long tail of discourse connectives would be out of the question, because it is precisely those less familiar expressions — which people encounter rarely and have weaker intuitions about — that would benefit the most from a rewrite. Two other examples, also cited in Webber and Joshi (2012), are automatic assessment of student essays (Burstein and Chodorow, 2010), and summarization (Thione et al., 2004). In student essays we encourage clear argumentative structure and rich vocabulary; failing to recognize that in an automatic system would not qualify as fair evaluation. And summarization is often performed over news wire, which, as shown in the PDTB, has a high per-article incidence of connectives not in top 10. Additionally, some low-frequency connectives like “ultimately” and “in particular” are strong cues for text selection.

Another reason to suspect that low-frequency connectives are important comes from an observation about the distribution of connectives in biomedical text. Ramesh and Yu (2010) report an overlap of only 44% between the connectives found in the The Biomedical Discourse Relation Bank (Prasad et al., 2011), a 24 article subset of the GENIA corpus (Kim et al., 2003), and the PDTB. The intersection contains high-frequency connectives, such as “and”, “however,” “also,” and “so”. Connectives specific to the biomedical domain include “followed by,” “due to,” and “in order to”, and the authors speculate that the unique connectives encode important domain specific knowledge.

## 3 Experiments

Our experiments are designed to shed light on three aspects of discourse connective disambiguation: 1) error distribution wrt. connective type; uneven performance builds a strong case for averaging over connective types instead of averaging over data points; 2) performance loss in the absence of an oracle; and 3) performance of simple model based on cheaper and more reliable annotations.

We experiment with three different feature sets,

all of which model syntactical aspects of the discourse connective.

The P&N and Lin feature sets are chosen to represent state-of-the-art. The high accuracy of P&N at 96.29% is frequently cited as an encouraging result, see Huang and Chen (2011; Alsaif and Markert (2011; Tonelli and Cabrio (2012; Zhou et al. (2010). Besides discourse parsing P&N has been used for tasks as diverse as measuring text coherence (Lin et al., 2011) and improving machine translation (Meyer and Popescu-Belis, 2012). The POS+LEX feature set is proposed as an alternative model. The baseline always predicts the majority class.

**P&N** This feature set derives from parse trees and replicates the features of Pitler and Nenkova (2009). Starting from the potential discourse connective, the features include the highest category in the tree subsuming only the connective called the self-category, the parent of that category, the left sibling of the self-category, and the right sibling of the self-category. A feature fires when the right sibling contains a VP, and another if there is a trace node below the right sibling. Note that the trace feature will never fire outside of the gold parse setting since state-of-the-art parsers do not predict trace nodes.

Importantly, there is a feature for the identity of the connective and interaction features between the connective and the syntactical features in effect allowing the model to fit parameters specific to each connective. Furthermore, combinations of the syntactical features are allowed, but they cannot be connective-specific.

**Lin** The feature set augments P&N with part-of-speech and string features for the tokens adjacent to the connective, as well as the part-of-speech of the connective itself. The part-of-speech features for the adjacent tokens interact with the part-of-speech of the connective, and the string features interact with the indicator feature for the connective. It also adds a syntax feature: the path to the root of the parse tree.

**POS+LEX** The simple feature set builds on part-of-speech tags and tokens. Part-of-speech tags are captured using a window of two tokens around the marker, and the lexical features are the same as Lin. Like P&N there is a feature for the identity of the connective as well as interaction

Model	Micro		Macro	
	Oracle	Pred.	Oracle	Pred.
Baseline	72.7	72.7	53.9	53.9
P&N	93.0	90.7	85.3	80.7
Lin	95.2	92.9	86.7	83.6
POS+LEX	89.7	89.7	82.5	83.5

Table 1: Comparing  $F_1$  score on oracle and predicted features using macro and micro averaging. A Wilcoxon signed rank test shows that the macro-averaged difference between POS+LEX and Lin10 using predicted features is not significant at  $p < 0.01$ .

features between the identity feature and other features.

In keeping with Pitler and Nenkova (2009) our learner is a maximum entropy classifier trained on sections 2-22 of the WSJ using ten-fold cross-validation.

### 3.1 Parsing Wall Street Journal

To obtain a version of the WSJ corpus containing fully predicted parses we use the Stanford Parser<sup>1</sup> training a separate model for each section. To parse a specific section we train on everything but that section (e.g. for parsing section 5 the training set is section 0-4 and 6-24). Average  $F_1$  on all sections is 85.87%. Although the very best state-of-the-art parsers<sup>2</sup> report  $F_1$  of above 90%, our parsing score greatly exceeds typical performance on real-life data, which is almost always out-of-domain with respect to 1980s WSJ. Thus this setting still compares favourably to performance in the wild.

## 4 Results

A summary of the results is found in Table 1. For a subset of frequent and less frequent connectives, Table 2 lists individual  $F_1$  scores. In all of the feature sets we see a marked drop moving from micro-average (average over instances) to macro-average (average over connective types)—P&N, for instance, goes from 93.0% to 85.3%. This shows that the scores of less frequent connectives are somewhat lower than frequent ones. When

<sup>1</sup><http://nlp.stanford.edu/software/lex-parser.shtml>, 2012-11-12 release with the 'goodPCFG' standard settings

<sup>2</sup>[http://aclweb.org/aclwiki/index.php?title=Parsing\\_\(State\\_of\\_the\\_art\)](http://aclweb.org/aclwiki/index.php?title=Parsing_(State_of_the_art))

	Oracle		Pred.		Disc.
	Lin	P+L	Lin	P+L	
but	98.6	96.1	97.6	96.1	78.9
and	94.9	77.0	89.0	77.0	14.7
also	97.0	97.3	97.5	97.2	93.5
if	93.4	93.1	92.3	93.0	82.6
when	89.9	88.5	89.3	88.4	65.5
because	99.5	99.4	99.4	99.5	63.4
while	97.6	97.7	97.5	97.4	91.9
as	89.8	63.1	78.1	63.0	13.0
after	93.7	74.0	87.9	72.9	42.4
however	98.7	98.4	98.5	98.4	95.7
...					
ultimately	43.2	30.3	36.4	29.4	37.5
rather	84.8	83.9	80.0	83.9	8.2
in other words	97.1	94.4	91.4	94.4	89.5
as if	84.8	84.8	71.0	88.2	66.7
earlier	76.9	66.7	74.1	69.6	2.1
meantime	80.0	76.5	82.4	80.0	71.4
in particular	89.7	85.7	85.7	80.0	48.4
in contrast	100.0	100.0	100.0	100.0	50.0
thereby	95.7	95.7	100.0	95.7	100.0
...					

Table 2:  $F_1$  score per connective. The table is sorted by the number of actual discourse connectives in the PDTB. After the break the table continues from position 50. The last column gives the percentage of discourse connectives.

features are derived from predicted parses performance also fall, from 93.0% to 90.7% with micro-average, and even more dramatically with macro-average, where it goes from 85.3% to 80.8%. Given that we are interested in real life performance this last figure is the most interesting.

## 5 Discussion

In NLP applications we cannot assume the existence of oracles providing us with gold-standard features. Often switching to predicted features introduces greater uncertainty. If the parser often confuses two non-terminals that are important for connective disambiguation we lose predictive power. Thus, on the P&N model, the average conditional entropy per feature given the class (how surprising the feature is when we know the answer) increases by 8.8% when the oracle is unavailable. In contrast there is almost no difference between the conditional entropy of the POS model with oracle features and without, indicating that the errors made by the tagger are not confusing in the disambiguation task.

Predicted parse features are associated with uncertainty even when used in combination with words and part of speech. Comparing the number

of times the Lin model changes an incorrect prediction of POS+LEX to a correct one and the number of times it introduces a new error by changing a correct prediction to an incorrect one, we observe that corrections almost always come with a substantial number of new errors. In fact, 58 connectives have at least as many new errors as corrections.

Predicted parse features also contribute to feature sparsity, because of the greater variability of automatic parses. On the other hand, they are more expressive than part of speech, and in the example below, where only Lin correctly identifies 'and' as a discourse connective, part of speech simply does not contain enough information.

“A whole day goes by **and** no one even knows they're alive.

## 6 Related work

Atterer and Schütze (2007) present similar experiments for prepositional phrase attachment showing that approaches assuming gold-standard features suffer a great deal when they are evaluated on predicted features. Spitkovsky et al. (2011) also caution against the use of gold-standard features, arguing that for unsupervised dependency parsing using induced parts of speech is superior to relying on gold-standard part-of-speech tags.

This work also relates to Manning (2011) who point out that even though part-of-speech tagging accuracy is above 97% the remaining errors are not randomly distributed but in fact occur in just the cases we care most about.

## 7 Conclusion

Discourse connective disambiguation is an important subtask of discourse parsing. We show that when realistic evaluation is adopted — averaging over connective types and not relying on oracle features — performance drops markedly. This suggests that more work on the task is needed. Moreover, we show that in realistic evaluation a simple feature model using part-of-speech tags and words performs just as well as a much more complex state-of-the-art model.

## Acknowledgements

We wish to thank the ESICT project for partly funding this work. The ESICT project is supported by the Danish Council for Strategic Research.

## References

- Amal Alsaif and Katja Markert. 2011. Modelling discourse relations for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 736–747, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michaela Atterer and Hinrich Schütze. 2007. Prepositional phrase attachment without oracles. *Computational Linguistics*, 33(4):469–476.
- Jill Burstein and Martin Chodorow. 2010. Progress and New Directions in Technology for Automated Essay Evaluation. In R. Kaplan, editor, *The Oxford Handbook of Applied Linguistics, 2nd Edition*, pages 487–497. Oxford University Press.
- Sucheta Ghosh. 2012. *End-to-End Discourse Parse using Cascaded Structured Prediction*. Phd thesis, University of Trento.
- Hen-Hsen Huang and Hsin-Hsi Chen. 2011. Chinese discourse relation recognition. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 1442–1446.
- J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for biotextmining. *Bioinformatics*, 19(Suppl 1):i180–i182, July.
- Ziheng Lin, Hwee Tou Ng, and Min-yen Kan. 2010. A PDTB-Styled End-to-End Discourse Parser. Technical Report 2004, School of Computing, National University of Singapore, Singapore.
- Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. pages 997–1006, June.
- Christopher D. Manning. 2011. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In AlexanderF. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing SE - 14*, volume 6608 of *Lecture Notes in Computer Science*, pages 171–189. Springer Berlin Heidelberg.
- Thomas Meyer and Andrei Popescu-Belis. 2012. Using Sense-labeled Discourse Connectives for Statistical Machine Translation. In *Proceedings of the Joint Workshop on Exploiting Synergies Between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, EACL 2012, pages 129–138, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort '09*, pages 13–16, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of LREC*.
- Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. 2011. The biomedical discourse relation bank. *BMC Bioinformatics*, 12(1):188.
- Balaji Polepalli Ramesh and Hong Yu. 2010. Identifying discourse connectives in biomedical text. In *AMIA Annual Symposium Proceedings*, volume 2010, page 657. American Medical Informatics Association.
- Valentin I Spitzkovsky, Hiyan Alshawi, Angel X Chang, and Daniel Jurafsky. 2011. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1281–1290. Association for Computational Linguistics.
- Gian Lorenzo Thione, Martin Van Den Berg, Livia Polanyi, and Chris Culy. 2004. Hybrid text summarization: Combining external relevance measures with structural analysis. In *Proceedings ACL Workshop Text Summarization Branches Out. Barcelona*.
- Sara Tonelli and Elena Cabrio. 2012. Hunting for Entailing Pairs in the Penn Discourse Treebank. In *Proceedings of COLING 2012*, pages 2653–2668, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Bonnie Webber and Aravind Joshi. 2012. Discourse Structure and Computation: Past, Present and Future. In *Association for Computational Linguistics*, page 42.
- Zhi Min Zhou, Man Lan, Zheng Yu Niu, Yu Xu, and Jian Su. 2010. The effects of discourse connectives prediction on implicit discourse relation recognition. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL '10*, pages 139–146, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Updating Rare Term Vector Replacement

Tobias Berka<sup>†</sup>

<sup>†</sup>Department of Computer Sciences  
University of Salzburg  
{tberka, marian}@cosy.sbg.ac.at

Marian Vajteršic<sup>†‡</sup>

<sup>‡</sup>Department of Informatics  
Mathematical Institute  
Slovak Academy of Sciences

## Abstract

Rare term vector replacement (RTVR) is a novel technique for dimensionality reduction. In this paper, we introduce an updating algorithm for RTVR. It is capable of updating both the projection matrix for the reduction and the reduced corpus matrix directly, without having to recompute the expensive projection operation. We introduce an effective batch updating algorithm, and present performance measurements on a subset of the Reuters newswire corpus that show that a 12.5% to 50% split of the documents into corpus and update vectors leads to a three to four fold speed-up over a complete rebuild. Thus, we have enabled optimized updating for rare term vector replacement.

## 1 Introduction

Rare term vector replacement (RTVR) is a recently developed linear dimensionality reduction technique for term frequency vectors (Berka and Vajteršic, 2011). It is easily and rapidly computed, patent-free, and produces a semantically meaningful multivariate space with a significantly reduced dimensionality. Furthermore, the method has been parallelized for data and task parallelism (Berka and Vajteršic, 2013).

The construction of this representation is based on document-scope term cooccurrences. Rare terms that occur only in  $\delta$  documents or fewer are eliminated by replacing them with the average vectors of the documents that contain them. The replacement vectors of the rare terms are added to the original document vectors. Then, all rows for rare terms are dropped from all vectors. Only the common terms remain of the original term frequency vector space. By Zipf's law (Powers, 1998), we know that this operation will eliminate

a high number of terms and lead to a highly condensed representation. The performance of the replacement now depends on the applicability of the *cluster hypothesis* (Raiber and Kurland, 2012; Rijsbergen, 1979), i.e., that documents in close proximity are semantically related. If the cluster hypothesis holds, then the centroid of the containing documents will act as a succinct representation of all documents containing the replacement term.

The most prominent techniques for dimensionality reduction of text data in published literature are latent semantic indexing (LSI), see (Deerwester et al., 1990) and the related COV approach (Kobayashi et al., 2002). These two methods are applications of a principal component analysis to text using unbiased and biased correlation measures. Factor analysis based on the SVD applied to automated indexing has been reported as probabilistic latent semantic analysis (PLSA) (Hofmann, 1999). Updating operations for LSI are well understood (Zha and Simon, 1999), and are also being developed for PLSA (Bassiou and Kotropoulos, 2011), or other dimensionality reduction methods such as the kernel PCA (Mastronardi et al., 2010).

Through the connection between PLSA and Latent Dirichlet Allocation (LDA) (Girolami and Kabán, 2003), topic models are also related to dimensionality reduction. Positive matrix factorization methods are also used in various text analysis tasks (Zhang, 2010). Structurally, the generalized vector space model (GVSM) (Wong et al., 1985), is similar to RTVR because of the construction of index vectors by linear combination. The random index vector representation (Kanerva et al., 2000) is also based on cooccurrences, but operates on random initial vectors. Random projections can be used to further accelerate it (Sakai and Imiya, 2009), but this approach should be seen as complementary because it can be applied to other methods as well.

Our contribution is the following. In this paper, we rigorously define an algorithm for updating rare term vector replacement. Using our approach, both the replacement vectors and the reduced corpus matrix are updated directly. It is not necessary to explicitly recompute the projection into the reduced-dimensional space.

The remainder of this paper is structured as thus. Our main contribution is the updating algorithm in Section 2. Section 3 contains a theoretic and empirical performance evaluation. Lastly, we summarize our findings in Section 4.

## 2 Updating RTVR

Updating RTVR has to support the three basic operations of content management: (1) adding new documents, (2) changing existing documents, and (3) deleting obsolete documents.

The replacement vectors are weighted centroids, or weighted average vectors, of the document vectors containing their terms. The reduced document vectors are also linear combinations of the truncated original document vectors and the replacement vectors. At its core, the update algorithm is therefore a running average computation. We will use the mathematical notation summarized in Table 1 to describe the algorithm.

A key complication lies in the fact that the occurrence counts of the terms change. This means that some rare terms may become common terms, and therefore become part of the reduced-dimensional, projected term space. Dually, some common terms may become rare terms, and drop out of the projected space. We will refer to these terms as *promoted* and *demoted terms*  $P$  and  $Q$ . For demoted terms, we need to compute the replacement vectors from scratch during the update. But for all rare terms that are involved in an update, in the old or new vector, we need to change the replacement vector. These terms are called *affected (rare) terms*  $A_T$ .

We can represent changing a document by a tuple  $(i, v)$  containing a corpus matrix column  $i \in \{1, \dots, n\}$  and a new term frequency vector  $v \in \mathbb{R}^{m'}$ , where  $m'$  is the new number of terms. The other two updating operations can be cast into the same form by introducing two abstract symbols. We let  $\nu$  denote the pseudo-column for adding new documents, i.e.,  $(\nu, d_{n+1})$  denotes a new document that will be added as a new column of the corpus matrix. For deletions, we let  $\epsilon$  denote

$T$	set of terms
$D$	set of documents
$m$	number of terms
$n$	number of documents
$C$	corpus matrix
$\mathcal{D}(t_i)$	set of documents containing $t_i$
$\mathcal{T}(d_j)$	non-zero terms in document $d_j$
$N_i$	occurrence count for term $t_i$
$\delta$	occurrence count threshold
$E$	set of rare terms
$\tau_E$	vector truncation removing indices in $E$
$k$	reduced dimensionality
$\pi$	index permutation mapping common features to reduced feature indices
$R$	replacement vectors
$\lambda$	normalizing factors
$\hat{C}$	reduced corpus
$U$	bulk update
$(i, v)$	update $v$ for $d_i$
$(\nu, v)$	insertion of document vector $v$
$(i, \epsilon)$	deletion of document $d_i$
$\text{old}(i, v)$	old vector for $i$ (or zero)
$\mathcal{T}_o(i, v)$	terms in the old vector
$\text{new}(i, v)$	new vector for $i$ (or zero)
$\mathcal{T}_n(i, v)$	terms in the new vector
$\mathcal{T}(i, v)$	terms in both vectors
$\mathcal{T}(U)$	all terms $\bigcup_{(i,v) \in U} \mathcal{T}(i, v)$ in $U$
$N'$	new occurrence counts
$E'$	new rare terms
$k'$	new reduced dimensionality
$\pi'$	new index permutation
$P$	promoted terms $\{t \in E \mid t \notin E'\}$
$Q$	demoted terms $\{t \in E' \mid t \notin E\}$
$A_T$	affected terms $\mathcal{T}(U) \setminus (P \cup Q)$
$\sigma$	index permutation to remove demoted terms
$e_i$	$i$ -th standard base

Table 1: Mathematical Notation

an empty pseudo-vector for the deletion of an old document, i.e.,  $(i, \epsilon)$  signifies the deletion of the document in column  $i$ .

We associate every update  $u = (i, v) \in U$  with two term frequency vectors. If the update is not an add document request, i.e.,  $i \neq \nu$ , it is associated with an old document vector  $\text{old}(i, v) = C_{1:m,i}$ . If it is not a deletion, i.e.,  $v \neq \epsilon$ , it is associated with a new document vector  $\text{new}(i, v) = v$ . We



---

**Algorithm 1:** Preparing the Update

---

```
delete or append terms in  $T, C, N, R, \lambda$ ;  
 $N' := N$ ;  $E' := E$ ;  
for  $u \in U$  do  
  Used := Used  $\cup$   $\mathcal{T}(u)$ ;  
  for  $t_i \in (\mathcal{T}_o(u) \setminus \mathcal{T}_n(u))$  do  $N'_i--$ ;  
  for  $t_i \in (\mathcal{T}_n(u) \setminus \mathcal{T}_o(u))$  do  $N'_i++$ ;  
for  $t_i \in T$  do  
  if  $(N'_i > \delta) \wedge t_i \in E$  then  
     $P := P \cup \{t_i\}$ ;  $E' := E' \setminus \{t_i\}$ ;  
  else if  $(N'_i \leq \delta) \wedge t_i \notin E$  then  
     $Q := Q \cup \{t_i\}$ ;  $E' := E' \cup \{t_i\}$ ;  
  else if  $(N'_i \leq \delta) \wedge t_i \in \text{Used}$  then  
     $A_T := A_T \cup \{t_i\}$ ;  
 $k'' := k - \|Q\|$ ;  $k' := k'' + \|P\|$ ;  
 $\sigma := 1_{k'}$ ;  $j := 1$ ;  $l := k'' + 1$ ;  
for  $t_i \in T$  do  
  if  $t_i \in P$  then  $\pi'(i) := l++$ ;  
  else if  $t_i \in A_T$  then  
     $\sigma(j) := \pi(i)$ ;  
     $\pi'(i) := j++$ ;  
  else  $\pi'(i) := -1$ ;
```

---

will need to identify the terms in the old vector  $\mathcal{T}_o(i, v)$ , in the new vector  $\mathcal{T}_n(i, v)$ , and the joint set  $\mathcal{T}(i, v)$ . All terms in the old and new vectors for the entire update  $U$  is defined as  $\mathcal{T}(U)$ .

Our updating algorithm proceeds in three phases: (1) preparing the update, (2) downdating the reduced corpus and updating the replacement vectors, and (3) updating the reduced corpus. In Algorithm 1, we analyze a batch update and prepare the required sets of features and an index permutation  $\sigma$  to compact the reduced space.

Let us assume that we have a procedure  $\text{Compact}(A, m', n, \sigma)$ , which applies the permutation  $\sigma$  to the row indices of a matrix. If we have a matrix  $A \in \mathbb{R}^{m \times n}$  and compute  $A' := \text{Compact}(A, m', n, \sigma) \in \mathbb{R}^{m' \times n}$ , it holds that  $A'_{i,j} = A_{\sigma(i),j}$ . Let us further assume that truncation of the new elimination terms  $\tau_{E'}$  respects the new index order established with  $\sigma$ , i.e., the implementation uses the global index permutation  $\pi'$  mapping all new features to indices in  $\{1, \dots, k'\}$ .

We downdate the reduced corpus and update the replacement vectors with Algorithm 2. For existing documents, we downdate the reduced corpus by subtracting any terms that will change in the course of the update in Line 1. We then update

the replacement vectors by adding the new document vector to any affected or demoted features in Line 2. Documents that do not change contribute to the construction of replacement vectors for demoted terms, which need to be built from scratch, in Line 3. These have to be inserted into rows  $k''$  to  $k'$  of the reduced space, as done in Line 4. We add in any new documents in Line 5 and normalize the resulting replacement vectors.

Algorithm 3 updates the reduced vectors by adding all replacement vectors that have changed in Line 1. The exception are the promoted terms on rows  $k''$  to  $k'$ , which must be added for all rare terms that were otherwise unaffected by the update in Line 2. We then handle all deletions and insertions in Lines 3 and 4.

### 3 Performance Evaluation

Regarding the asymptotic complexity of our updating algorithm, we note the following. Assuming amortized constant time for set testing and insertion (Sedgewick, 2002), that the update is smaller than the corpus, i.e.,  $\|U\| < \|D\|$ , and that the number of documents is greater than the number of terms, i.e.,  $n > m$ , the update algorithm can be executed with a complexity of  $O((\|D\| + \|U\|) \overline{nnz} k' + mk')$ , where  $\overline{nnz}$  is the expected number of non-zero elements in any document vector.

Since the performance is heavily dependent on the actual distribution of the non-zero elements, the observed performance may differ somewhat from this formal analysis. We have conducted performance measurements using an Intel i5-2557 with 4 GB or RAM running Max OS X 10.7.5. We have used the first 23,149 documents with 47,236 terms of the Reuters Corpus Volume I, version 2, in the pre-vectorized form (Lewis et al., 2004).

We randomly selected between 12.5% and 50% of all vectors for the batch update, using the remaining documents for the initial build. Table 2 summarizes our performance measurements averaged across ten runs per row.

The results clearly show that the updating algorithm outperforms a complete rebuild with the original construction algorithm by three-fold to four-fold performance improvement. Because smaller updates require less processing in the updating, and the workload for the rebuild remains the same, smaller batches have a larger speed-up than smaller batches.

---

**Algorithm 2:** Downdating the Reduced Corpus and Updating the Replacement Vectors

---

```

for  $t_i \in A_T$  do  $R_{*,i}^* = \lambda_i$ ;
 $R' = \begin{bmatrix} \text{Compact}(R, k'', m, \sigma) \\ 0 \end{bmatrix} \in \mathbb{R}^{k' \times m}$ ;
for  $t_i \in Q$  do  $R'_{*,i} = 0$ ;
for  $d_j \in D$  do
1 for  $t_i \in \mathcal{T}(d_j)$  do
    if  $(j, d'_j) \in U \wedge t_i \in A_T$  then
        for  $l \in \{1, \dots, k''\}$  do
             $R'_{l,i} = C_{i,j} \tau_E(C_{*,j})_{\sigma(l)}$ ;
             $\lambda_i = |C_{i,j}|$ ;
        if  $(j, d'_j) \notin U \wedge t_i \in A_T \cup P$  then
             $\hat{C}_{*,j} = C_{i,j} R_{*,i}$ ;
2 if  $(j, d'_j) \in U$  then
    for  $t_i \in \{t_i \in T \mid (d'_j)_i \neq 0\}$  do
        if  $t_i \in Q \cup A_T$  then
            for  $l \in \{1, \dots, k'\}$  do
                 $R'_{l,i} += (d'_j)_i \tau_{E'}(d'_j)_l$ ;
                 $\lambda_i += |(d'_j)_i|$ ;
3 else for  $t_i \in \mathcal{T}(d_j)$  do
    if  $t_i \in Q$  then
        for  $l \in \{1, \dots, k'\}$  do
             $R'_{l,i} += C_{i,j} \tau_{E'}(C_{*,j})_l$ ;
             $\lambda_i += |C_{i,j}|$ ;
    else if  $t \in E'$  then
4 for  $l \in \{k'', \dots, k'\}$  do
         $R'_{l,i} += C_{i,j} \tau_{E'}(C_{*,j})_l$ ;
        if  $(\mathcal{T}(d_j) \cap E') \subseteq P$  then
             $\lambda_i += |C_{i,j}|$ ;
5 for  $(\nu, v) \in U$  do
    for  $t_i \in \mathcal{T}_n(\nu, v) \cap E'$  do
        for  $l \in \{1, \dots, k'\}$  do
             $R'_{l,i} += C_{i,j} \tau_{E'}(v)_l$ ;
             $\lambda_i += |v_j|$ ;
for  $t_i \in E'$  do
    if  $t_i \in P$  then  $R'_{*,i} = \tau_{E'}(e_i)$ ;
    else  $R'_{*,i} = \lambda_i$ ;

```

---



---

**Algorithm 3:** Updating the Reduced Corpus

---

```

 $\hat{C}' = \begin{bmatrix} \text{Compact}(\hat{C}, k'', n, \sigma) \\ 0 \end{bmatrix} \in \mathbb{R}^{k' \times n}$ ;
Old =  $E' \setminus (P \cup A_T \cup Q)$ ;
1 for  $d_j \in D$  do
    if  $(j, d'_j) \in U$  then
         $\hat{C}'_{*,j} = \tau_{E'}(d'_j) + \sum_{t_i \in E'} (d'_j)_i R'_{*,i}$ ;
    else
         $\hat{C}'_{*,j} += \sum_{t_i \in P} C_{i,j} \tau_{E'}(e_i)$ ;
         $\hat{C}'_{*,j} += \sum_{t_i \in Q \cup A_T} C_{i,j} R'_{*,i}$ ;
2  $\hat{C}'_{k'':k',j} += \sum_{t_i \in \text{Old}} C_{i,j} R'_{k'':k',i}$ ;
3 if  $(j, \epsilon) \in U$  then
     $\hat{C}' = \begin{bmatrix} \hat{C}'_{*,1:j-1} & \hat{C}'_{*,j+1:n} \end{bmatrix}$ ;
     $n --$ ;
4 for  $(\nu, v) \in U$  do
     $v' = \tau_{E'}(v) + \sum_{t_i \in E'} v_i R'_{*,i}$ ;
     $\hat{C}' = \begin{bmatrix} \hat{C}' & v' \end{bmatrix}$ ;
     $n ++$ ;

```

---

Size	$t_R$	$t_I$	$t_U$	$S$
12.5%	51.41	43.59	11.44	4.49
25.0%	51.41	40.73	13.81	3.72
37.5%	51.41	34.57	15.15	3.39
50.0%	51.41	28.52	16.12	3.18

Table 2: Performance evaluation (rebuild time  $t_R$ , build time  $t_I$ , update time  $t_U$  and speed-up  $S$ ).

## 4 Summary & Conclusions

In this paper, we have introduced an algorithm for updating rare term vector replacement. Our empirical performance evaluation demonstrates that batch updating is faster than a complete rebuild by a factor of three to four for our experiments. In our future research, we intend to develop hybrid updating algorithms similar to (Tougas and Spiteri, 2008). These algorithms initially compute fast, approximate updates, which are only later replaced by exact updates for efficiency. The final PCA of the augmented corpus  $\hat{C}$  reported in (Berka and Vajteršic, 2011) remains an open problem in updating RTVR.

## Acknowledgements

We acknowledge the support of the Slovak Ministry of Education and Slovak Academy of Sciences under VEGA grant no. 2/0003/11.

## References

- N. Bassiou and C. Kotropoulos. 2011. RPLSA: A Novel Updating Scheme for Probabilistic Latent Semantic Analysis. *Comput. Speech Lang.*, 25(4):741–760.
- T. Berka and M. Vajteršič. 2011. Dimensionality Reduction for Information Retrieval using Vector Replacement of Rare Terms. In *Proc. TMW*.
- T. Berka and M. Vajteršič. 2013. Parallel Rare Term Vector Replacement: Fast and Effective Dimensionality Reduction for Text. *J. Parallel Distr. Com.*, 73(3):341–351.
- S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by Latent Semantic Analysis. *JASIS*, 41(6):391–407.
- M. Girolami and A. Kabán. 2003. On an Equivalence Between PLSI and LDA. In *Proc. SIGIR*, pages 433–434, USA. ACM.
- T. Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proc. SIGIR*, pages 50–57, USA. ACM.
- P. Kanerva, J. Kristoferson, and A. Holst. 2000. Random Indexing of Text Samples for Latent Semantic Analysis. In *Proc. CogSci*, pages 103–106. Erlbaum.
- M. Kobayashi, M. Aono, H. Takeuchi, and H. Samukawa. 2002. Matrix Computations for Information Retrieval and Major and Outlier Cluster Detection. *J. Comput. Appl. Math.*, 149(1):119 – 129.
- D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *JMLR*, 5:361–397.
- N. Mastronardi, E. E. Tyrtshnikov, and P. Van Dooren. 2010. A Fast Algorithm for Updating and Downsizing the Dominant Kernel Principal Components. *SIAM J. Matrix Anal. Appl.*, 31(5):2376–2399.
- D. M. W. Powers. 1998. Applications and Explanations of Zipf’s Law. In *Proc. NeMLaP3/CoNLL*, pages 151–160, USA. ACL.
- F. Raiber and O. Kurland. 2012. Exploring the Cluster Hypothesis, and Cluster-Based Retrieval, Over the Web. In *Proc. CIKM*, pages 2507–2510, USA. ACM.
- C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London.
- T. Sakai and A. Imiya. 2009. Fast Spectral Clustering with Random Projection and Sampling. In *Machine Learning and Data Mining in Pattern Recognition*, LLNCS, pages 372–384. Springer.
- R. Sedgewick. 2002. *Algorithms in C++*. Addison Wesley, 3rd edition.
- J. E. Tougas and R. J. Spiteri. 2008. Two Uses for Updating the Partial Singular Value Decomposition in Latent Semantic Indexing. *Appl. Numer. Math.*, 58(4):499–510.
- S. K. M. Wong, W. Ziarko, and P. C. N. Wong. 1985. Generalized Vector Spaces Model in Information Retrieval. In *Proc. SIGIR*, pages 18–25, USA. ACM.
- H. Zha and H. D. Simon. 1999. On Updating Problems in Latent Semantic Indexing. *SIAM J. Sci. Comput.*, 21(2):782–791.
- Z.-Y. Zhang. 2010. Survey on the Variations and Applications of Nonnegative Matrix Factorization Variations of NMF. *Operations Research*, pages 317–323.

# Statistical Morphological Analyzer for Hindi

Deepak Kumar Malladi and Prashanth Mannem

Language Technologies Research Center

International Institute of Information Technology

Hyderabad, AP, India - 500032

{deepak.malladi, prashanth}@research.iiit.ac.in

## Abstract

Morphology is the study of internal structure of words and is an essential early step in many NLP applications such as parsing and machine translation. Researchers working in Hindi NLP have either used the widely popular paradigm based analyzer (PBA) or extensions of it. In this work, we undertook a comprehensive evaluation of PBA using the data from the Hindi Treebank (HTB) and presented a new morphological analyzer trained on the HTB. Our morphological analyzer has better coverage and accuracy when compared to the existing analyzers for Hindi. An oracle system that takes the best values from the PBA's output achieves only 63.41% for lemma, gender, number, person and case. Our statistical analyzer has an accuracy of 84.16% for these morphological attributes when evaluated on the test section of the Hindi Treebank.

## 1 Introduction

Morphological analysis is the task of analyzing the structure of morphemes in a word and is generally a prelude to further complex NLP tasks such as parsing, machine translation, semantic analysis etc. These tasks need an analysis of the words in the sentence in terms of lemma, affixes, parts of speech (POS) etc.

Hindi is a morphologically rich language with a relatively free word order. Previous efforts in Hindi morphological analysis concentrated on building rule-based systems that give all the possible analyses for a word form irrespective of its context in the sentence. The paradigm based analyzer (PBA) by Bharati et al. (1995) is one of the most widely used applications among researchers in the Indian NLP community. In paradigm

based analysis, words are grouped into a set of paradigms depending on the inflections they take. Each paradigm has a set of add-delete rules to account for its inflections and words belonging to a paradigm take the same inflectional forms. Given a word, the PBA identifies the *lemma*, *coarse POS tag*, *gender*, *number*, *person*, *case marker*, *vibhakti*<sup>1</sup> and *TAM* (tense, aspect, modality). Being a rule-based system, the PBA takes a word as input and gives all the possible analyses as output. (Table 1 presents an example). It doesn't pick the correct analysis for a word in its sentential context.

Goyal and Lehal's analyser (2008), which is a re-implementation of the PBA with few extensions, has not done any comparative evaluation. Kanuparthi et al. (2012) built a derivational morphological analyzer for Hindi by introducing a layer over the PBA. It identifies 22 derivational suffixes which helps in providing derivational analysis for the word whose suffix matches with one of these 22 suffixes.

The large scale machine translation projects<sup>2</sup> that are currently under way in India use shallow parser built on PBA and an automatic POS tagger. The shallow parser prunes the morphological analyses from PBA to select the correct one using the POS tags from the tagger. Since it is based on PBA, it suffers from similar coverage issues for out of vocabulary (OOV) words.

The PBA, developed in 1995, has a limited vocabulary and has received only minor upgrades since then. Out of 17,666 unique words in the Hindi Treebank (HTB) released during the 2012 Hindi Parsing Shared Task (Sharma et al., 2012), the PBA does not have entries for 5,581 words (31.6%).

NLP for Hindi has suffered due to the lack of a

<sup>1</sup>Vibhakti is a Sanskrit grammatical term that encompasses post-positionals and case endings for nouns, as well as inflection and auxiliaries for verbs (Pedersen et al., 2004).

<sup>2</sup><http://sampark.iiit.ac.in/>

	L	G	N	P	C	T/V
	↓	↓	↓	↓	↓	↓
xeSa	xeSa	m	sg	3	d	0
(country)	xeSa	m	pl	3	d	0
	xeSa	m	sg	3	o	0
cAhie	cAha	any	sg	2h	-	ie
(want)	cAha	any	pl	2h	-	eM

L-lemma, G-gender, N-number, P-person  
C-case, T/V-TAM or Vibhakti

Table 1: Multiple analyses given by the PBA for the words xeSa and cAhie

high coverage automatic morphological analyzer. For example, the 2012 Hindi Parsing Shared Task (Sharma et al., 2012) held with COLING-2012 workshop had a gold-standard input track and an automatic input track, where the former had gold-standard morphological analysis, POS tags and chunks of a sentence as input and the automatic track had only the sentence along with automatic POS tags as input. The morphological information which is crucial for Hindi parsing was missing in the automatic track as the existing analyzer had limited coverage.

In this work, we present a statistical morphological analyzer for Hindi trained on HTB and compare it with PBA. The analyzer predicts the *lemma*, *gender*, *number*, *person* (GNP) and *case marker* for all the words in a given sentence by training separate models on the HTB for each of them. Other grammatical features such as TAM (tense, aspect, modality) and *vibhakti* are predicted using heuristics on fine grained POS tags of the input sentence. Our system has significantly better accuracy than analyzers based on PBA and is robust enough to produce analyses for OOV words.

## 2 Statistical Morphological Analyzer (SMA)

The output of a morphological analyzer depends on the language that it is developed for. Analyzers for English (Goldsmith, 2000) predict just the lemmas and affixes mainly because of its restricted agreement based on semantic features such as animacy and *natural* gender. But in Hindi, agreement depends on *lexical* features such as *grammatical* gender, number, person and case. Hence, it is crucial that Hindi analyzers predict these along with TAM and vibhakti which have been found to be useful for syntactic parsing (Ambati et al., 2010; Bharati et al., 2009a).

MorphFeature	Values
Gender	masculine, feminine, any, none
Number	singular, plural, any, none
Person	1, 1h, 2, 2h, 3, 3h, any, none
CaseMarker	direct, oblique, any, none

Table 2: Morph features and the values they take

Hindi has syntactic agreement (of GNP and case) of two kinds: modifier-head agreement and noun-verb agreement. Modifiers, including determiners, agree with their head noun in gender, number and case, and finite verbs agree with some noun in the sentence in gender, number and person (Kachru, 2006). Therefore, apart from lemma and POS tags, providing gender, number and person is also crucial for syntactic parsing.<sup>3</sup>

With the existing morph analyzer (PBA) performing poorly on OOV (unknown to PBA) words and the availability of an annotated treebank, we set out to build a high-coverage automatic Hindi morph analyzer by learning each of the seven morphological attributes separately from the Hindi Treebank. During this process, it was realized that vibhakti and TAM can be better predicted using heuristics on fine-grained POS tags than by training on the HTB.

In the rest of the section, we discuss the methods to predict each of the seven morphological attributes. Table 2 lists the values that each of the morph attributes take in HTB. The HTB consists of 15,102 sentences (334,287 words) annotated with morphological features, POS tags, chunks and dependency relations. In this work, we only use morph and POS information.

### 2.1 Lemma prediction

The PBA uses a large vocabulary along with paradigm tables consisting of add-delete rules to find the lemma of a given word. All possible add-delete rules are applied on a given word form and the resulting lemma is checked against the vocabulary to find if it is right or not. If no such lemma exists (for OOV words), it returns the word itself as the lemma.

While the gender, number and person of a word form varies according to the context (due to syntactic agreement with head words), there are very

<sup>3</sup>While nouns, pronouns and adjectives have both GNP and case associated with them, verbs only have GNP. TAM is valid only for verbs and vibhakti (post-position) is only associated with nouns and pronouns.

Analysis	Test Data - Overall(%)				Test Data - OOV of SMA(%)			
	Baseline	F-PBA	O-PBA	SMA	Baseline	F-PBA	O-PBA	SMA
L	71.12	83.10	86.69	95.70	78.10	82.08	82.48	85.82
G	37.43	72.98	79.59	95.43	60.22	43.07	44.06	79.09
N	52.87	72.22	80.50	94.90	69.60	44.53	47.56	89.12
P	45.59	74.33	84.13	95.77	78.30	52.51	53.89	94.39
C	29.31	58.24	81.20	94.62	43.60	31.40	47.36	87.40
V/T	65.40	53.05	59.65	97.04	58.31	33.58	34.56	96.04
L+C	16.46	48.84	72.06	90.67	32.52	28.50	44.66	75.33
L+V/T	54.78	44.57	51.71	92.93	53.56	31.73	32.72	82.65
G+N+P	23.05	61.10	73.81	89.42	47.49	35.75	39.58	71.31
G+N+P+C	9.72	45.73	70.87	85.56	21.04	20.91	35.95	64.64
L+G+N+P	20.27	53.29	66.28	85.88	44.72	34.63	38.46	62.34
L+G+N+P+C	8.57	38.25	63.41	<b>82.16</b>	19.33	19.92	34.89	<b>56.66</b>
L+G+N+P+C+V/T	1.25	32.53	42.80	<b>80.11</b>	4.02	14.51	18.67	<b>54.35</b>

L-lemma, G-gender, N-number, P-person, C-case, V/T-Vibhakti/TAM

Table 3: Accuracies of SMA compared with F-PBA, O-PBA and baseline systems.

few cases where a word form can have more than one lemma in a context. This makes lemma simpler to predict among the morphological features, provided there is access to a dictionary of all the word forms along with their lemmas. Unfortunately, such a large lemma dictionary doesn't exist.

In this work, we perceived lemma prediction from a machine translation perspective, with the characters in the input word form treated as the source sentence and those in the lemma as the target. The strings on source and target side are split into sequences of characters separated by space. The phrase based model (Koehn et al., 2007) in Moses is trained on the parallel data created from the training part of HTB. The translation model accounts for the changes in the affixes (sequence of characters) from word form to lemma whereas the language model accounts for which affixes go with which stems. In this perspective, the standard MT experiment of switching source and target to attain better accuracy would not apply since it is unreasonable to predict the word form from the lemma without taking the context into account.

## 2.2 Gender, Number, Person and Case Prediction

Unlike lemma prediction, we use a liblinear classifier (Fan et al., 2008) to build linear SVM classification models for GNP and case prediction.

Though knowing the syntactic head of a word helps in enforcing agreement (and thereby accu-

rately predicting the correct GNP), parsing is usually a higher level task and is not performed before morphological analysis. Hence, certain cases of GNP prediction are similar in nature to the standard chicken and egg problem.

The following features were tried out in building the models for gender, number, person and case prediction:

- Word
- Lexical category
- Last 3 characters
- Last 4 characters
- Next word
- Previous word
- Lemma
- Word Length
- Character N-grams of the word

## 2.3 Vibhakti and TAM

Vibhakti and TAM are helpful in identifying the *karaka*<sup>4</sup> dependency labels in HTB. While nouns and pronouns take vibhakti, verbs inflect for TAM. Both TAM and vibhakti occur immediately after the words in their respective word classes.

Instead of building statistical models for vibhakti and TAM prediction, we built a system that uses heuristics on POS tag sequences to predict the correct value. The POS tags of words following nouns, pronouns and verbs give an indication as to what the vibhakti/TAM are. Words with PSP (postposition) and NST (noun with spatial and temporal properties) tags are generally considered as the vibhakti for the preceding nouns and

<sup>4</sup>karakas are syntactico-semantic relations which are employed in Paninian framework (Begum et al., 2008; Bharati et al., 2009b)

Data	#Sentences	#Words
Training	12,041	268,096
Development	1,233	26,416
Test	1,828	39,775

Table 4: HTB statistics

pronouns. A postposition in HTB is annotated as PSP only if it is written separately (*usane/PRP* vs *usa/PRP ne/PSP*). For cases where the postposition is not written separately we rely on the treebank data to get the suffix. Similarly, words with VAUX tag form the TAM for the immediately preceding verb.

The PBA takes individual words as input and hence does not output the entire vibhakti or TAM of the word in the sentence. It only identifies these values for those words which have the information within the word form (e.g. *usakA he+Oblique*, *kiyA do+PAST*).

In the sentence,

```
rAma/NNP kA/PSP kiwAba/NN
cori/NN ho/VM sakawA/VAUX
hE/VAUX,
```

PBA identifies *rAma*'s vibhakti as *0* and *ho*'s TAM as *0*. Whereas in HTB, vibhakti and TAM of *rAma* and *ho* are annotated as *0\_kA* and *0\_saka+wA.hE* respectively. Our approach determines this information precisely.

### 3 Experiments and Results

The Hindi treebank released as part of the 2012 Hindi Parsing Shared Task is used to evaluate our models. All the models are tuned on development data and evaluated on test data. Table 4 shows the word counts of training, development and test sections of HTB.

Our approach to Hindi morphological analysis is based on handling each of the seven attributes (*lemma, gender, number, person, case, vibhakti* and *TAM*) separately. However, evaluation is performed on individual attributes as well as on the combined output. The models are compared with a baseline system and two versions of the PBA wherever relevant. The *baseline* system takes the word form itself as the lemma and selects the most frequent value for the rest of the attributes.

Since PBA is a rule-based analyzer which gives more than one analysis for words, we use two versions of it for comparison. The first system is

the oracle PBA (referred further as O-PBA) which uses an oracle to pick the *best* analysis from the list of all analyses given by the PBA. The second version of the PBA (F-PBA) picks the *first* analysis from the output as the correct analysis.

Table 3 presents the accuracies of four systems (baseline, F-PBA, O-PBA and SMA) in predicting the morphological attributes of all the words in the HTB's test data and also for OOV words of SMA (i.e. words that occur in the test section but not in training section of HTB)<sup>5</sup>. The accuracies are the percentages of words in the data with the correct analysis. It may be noted that our system (SMA) performs significantly better than the best analyses of PBA and the baseline system in all the experiments conducted.

The existing Hindi POS tagger<sup>6</sup> was found to be 95% accurate when evaluated on the entire HTB data. We got similar results when we had run the entire set of experiments using these automatic POS tags.

### 4 Conclusion and Future work

In conclusion, our paper presented a robust state-of-the-art statistical morphological analyzer for Hindi which outperforms previous analyzers by a considerable margin. A comprehensive evaluation was carried out for our system by comparing it with the existing analyzers. The analyzer we have developed achieved an accuracy of 82.03% for lemma, gender, number, person, case, vibhakti and TAM. Being a statistical model, it even analyzes OOV words thereby extending the coverage of the analyzer. We also evaluated the effect of morphological features (predicted by our system) on dependency parsing and found them to improve the parsing accuracy.

The agreement phenomenon in Hindi provides challenges in predicting gender, number and person of words in their sentential context. These can be better predicted if dependency relations are given as input. However, the standard natural language analysis pipeline forbids using parse information during morphological analysis.

This provides an opportunity to explore joint modelling of morphological analysis and syntactic parsing for Hindi. We plan to experiment this as part of our future work.

<sup>5</sup>OOV words for SMA need not be *out of vocabulary* for PBA's dictionaries.

<sup>6</sup>ilmt.iiit.ac.in

## References

- Bharat Ram Ambati, Samar Husain, Joakim Nivre, and Rajeev Sangal. 2010. On the role of morphosyntactic features in hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 94–102. Association for Computational Linguistics.
- Rafiya Begum, Samar Husain, Arun Dhvaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of IJCNLP*.
- Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural language processing: A Paninian perspective*. Prentice-Hall of India New Delhi.
- Akshar Bharati, Samar Husain, Meher Vijay, Kalyan Deepak, Dipti Misra Sharma, and Rajeev Sangal. 2009a. Constraint based hybrid approach to parsing indian languages. *Proc of PACLIC 23. Hong Kong*.
- Akshara Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begam, and Rajeev Sangal. 2009b. Anncorra: Treebanks for indian languages, guidelines for annotating hindi treebank.
- 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.
- John Goldsmith. 2000. Linguistica: An automatic morphological analyzer. In *Proceedings of 36th meeting of the Chicago Linguistic Society*.
- Vishal Goyal and G. Singh Lehal. 2008. Hindi morphological analyzer and generator. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 1156–1159. IEEE.
- Yamuna Kachru. 2006. *Hindi*, volume 12. John Benjamins Publishing Company.
- Nikhil Kanuparthi, Abhilash Inumella, and Dipti Misra Sharma. 2012. Hindi derivational morphological analyzer. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 10–16. 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, 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.
- Mark Pedersen, Domyenyk Eades, Samir K Amin, and Lakshmi Prakash. 2004. Relative clauses in hindi and arabic: A paninian dependency grammar analysis. *COLING 2004 Recent Advances in Dependency Grammar*, pages 9–16.
- Dipti Misra Sharma, Prashanth Mannem, Joseph Van-Genabith, Sobha Lalitha Devi, Radhika Mamidi, and Ranjani Parthasarathi, editors. 2012. *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*. Mumbai, India, December.



# Induction of Root and Pattern Lexicon for Unsupervised Morphological Analysis of Arabic

**Bilal Khaliq**

Dept of Informatics, University of Sussex  
Brighton BN1 9QJ, UK  
bk54@sussex.ac.uk

**John Carroll**

Dept of Informatics, University of Sussex  
Brighton BN1 9QJ, UK  
J.A.Carroll@sussex.ac.uk

## Abstract

We propose an unsupervised approach to learning non-concatenative morphology, which we apply to induce a lexicon of Arabic roots and pattern templates. The approach is based on the idea that roots and patterns may be revealed through mutually recursive scoring based on hypothesized pattern and root frequencies. After a further iterative refinement stage, morphological analysis with the induced lexicon achieves a root identification accuracy of over 94%. Our approach differs from previous work on unsupervised learning of Arabic morphology in that it is applicable to naturally-written, unvowelled text.

## 1 Introduction

Manual development of morphological analysis systems is expensive. It is impractical to develop morphological descriptions for more than a very small proportion of human languages. In recent years a number of approaches have been proposed that learn the morphology of a language from unannotated text. The Morpho Challenge and similar competitions have further motivated researchers to devise techniques for unsupervised learning of language.

Previous work in unsupervised morphology learning has mostly addressed concatenative morphology, in which surface word forms are sequentially separated or segmented into morpheme units. However, some languages (in particular Semitic languages) have another type of word formation in which morphemes combine in a non-concatenative manner, through the interdigitation of a root morpheme with an affix or pattern template. Unsupervised learning of non-concatenative morphology has received comparatively little attention.

In this paper we describe a conceptually simple yet effective unsupervised approach to learning non-concatenative morphology. We apply our approach to inducing an Arabic lexicon of trilateral roots and pattern templates. Lexicon acquisition is based on the idea that roots and affix patterns may be revealed by their converses, i.e. roots are identified from occurrences of patterns and conversely patterns are recognized from root frequencies. Subsequently, the lexicons are iteratively improved by refining the morpheme strengths computed in the previous step.

The paper is organized as follows. We survey previous related work (Section 2), and then give a brief introduction to Arabic root and pattern morphology (Section 3). We explain our basic technique for unsupervised lexicon induction in Section 4, followed by the refinement procedure (Section 5). Section 6 describes how the lexicon is used for morphological analysis. Finally, we present an evaluation (Section 7) and conclusions (Section 8).

## 2 Related Work

Beesley (1996) describes one of the first morphological analysis systems for Arabic, based on finite-state techniques with manually acquired lexicons and rules. This kind of approach, although potentially producing an efficient and accurate system, is expensive in time and linguistic expertise, and lacks robustness in terms of extendibility to word types not in the dictionary (Ahmed, 2000).

Darwish (2002) describes a semi-automatic technique that learns morphemes and induces rules for deriving stems using an existing dictionary of word and root pairs. It is an easy to build and fairly robust method of performing morphological analysis. Clark (2007) investigates semi-supervised learning using the

complex broken plural structure of Arabic as a test case. He employs memory-based algorithms, with the aim of gaining insights into human language acquisition.

Other researchers have applied statistical and information-theoretic approaches to unsupervised learning of morphology from raw (unannotated) text corpora. Goldsmith (2000, 2006) and Cruetz and Lagus (2005, 2007) use the Minimum Description Length (MDL) principle, considering input data to be ‘compressed’ into a morphologically analysed representation. An alternative perspective adopted by Schone and Jurafsky (2001) induces semantic relatedness between word pairs by Latent Semantic Indexing.

Most work on unsupervised learning of morphology has focused on concatenative morphology (Hammarström and Borin, 2011). Studies that have focussed on non-concatenative morphology include that of Rodriguez and Čavar (2005), who learn roots from artificially generated text using a number of orthographic heuristics, and then apply constraint-based learning to improve the quality of the roots. Xanthos (2008) deciphers roots and patterns from phonetic transcriptions of Arabic text, using MDL to refine the root and pattern structures.

Our work differs from these previous approaches in that (1) we learn intercalated morphology, identifying the root and transfixes/incomplete pattern for words, and (2) we start from ‘natural’ text without short vowels or diacritical markers.

### 3 Root and Pattern Morphology

Words in Arabic are formed through three morphological processes: (i) fusion of a root form and pattern template to derive a base word; (ii) affixation, including inflectional morphemes marking gender, plurality and/or tense, resulting in a stem; and (iii) possible attachment of a final layer of clitics. Our work addresses the first two of these processes.

As an example of word formation in Arabic, the word *ktAby* is formed from the root *Ktb* and the pattern *--A-y*, where *y* is an inflectional marker and *A* is the derivational infix marker for nouns.

During analysis, we decompose each word *w* into a set of tuples encoding all *k* possible combinations of a root (of at least 3 letters) and associated pattern (Eq. 1)

$$d(w) \rightarrow \{(r^x, p^x)\} \quad (\text{Eq. 1})$$

where *x* ranges from 1 to *k*. For example, the decomposition of the word *yErf* is shown in Figure 1.

$$yErf \rightarrow \left\{ \begin{array}{l} \langle y E r, \quad - - - f \rangle, \\ \langle y E f, \quad - - r - \rangle, \\ \langle y r f, \quad - E - - \rangle, \\ \langle E r f, \quad y - - - \rangle, \\ \langle y E r f, \quad - - - - \rangle \end{array} \right\}$$

Figure 1. Decomposition of a word into all possible combinations of roots and patterns.

## 4 Building Lexicons Using Contrastive Scoring

Based on the idea that roots and patterns may be revealed by their converses, we score a pattern based on the frequency of occurrence of the roots, and score a root according to the number of occurrences of patterns. We score each morpheme and then rescore it weighted by previous scores. Our technique resembles the *hubs and authorities* algorithm originally devised for rating Web pages (Kleinberg, 1999), which assigns to each Web page two scores: its hub value and its authority. These two values are updated in a similar mutually recursive manner as we describe for roots and patterns.

### 4.1 Frequency-Based Scoring

The initial scoring function is simple: firstly, we aggregate over the number of occurrences of a root radical sequence in a word  $w_i$ , for words  $i=1,2,\dots,N$  in the input dataset. The function for scoring each pattern in the target word,  $t$ , is given in equation (Eq. 2).

$$S(p_t^x) = \sum_{i=1}^N (1 | r_t^x = r_{w_i}^y) \quad (\text{Eq. 2})$$

The function for scoring the root,  $r_t^x$ , in each target word,  $t$ , with pattern,  $p_t^x$ , is given in equation (Eq. 3).

$$S(r_t^x) = \sum_{i=1}^N (1 | p_t^x = p_{w_i}^y) \quad (\text{Eq. 3})$$

We choose this as our baseline, to which we compare subsequent enhancements.

## 4.2 Scaling

Since pattern strength is computed based on root occurrence frequency and vice-versa, each pattern and root has a different score range due to the distinct distributions of patterns and roots. In order to make the scores comparable and contribute equally, we scale the scores in one lexicon with respect to the other.

We take the pattern lexicon as reference and scale each root,  $r_u$  ( $u=1,2,\dots,R$  entries in root lexicon) by the ratio of the maximum pattern score to the maximum root score:

$$SS(r_u) = S(r_u) \left( \times \frac{\max(S(p))}{\max(S(r))} \right) \quad (\text{Eq. 4})$$

## 4.3 Iterative Rescoring

Having obtained initial scores for the root and pattern lexicons, they are improved through an iterative rescoring process. We rescore each morpheme lexicon in a similar manner to equations (Eq. 2) and (Eq. 3), but weighted with the normalized score for each morpheme of previous scores. This is an iterative process starting with the initial score,  $S_0$ , calculated using frequency counts (as in section 4.1). Let  $S_j$  be the new score based on previous scores, and scaled scores,  $S_{j-1}$  and  $SS_{j-1}$ , respectively, for iterations  $j=0,1,2,\dots,n$ ,

$$S_j(r_t^x) = \sum_{i=1}^N \left( S_{j-1}(p_i^x) / \max(S_{j-1}) \mid p_i^x = p_{w_i}^y \right) \quad (\text{Eq. 5})$$

$$S_j(p_w^x) = \sum_{i=1}^N \left( SS_{j-1}(r_i^x) / \max(SS_{j-1}) \mid r_i^x = r_{w_i}^y \right) \quad (\text{Eq. 6})$$

Here we have normalized the score with respect to the maximum value for the reference pattern lexicon, thus keeping the magnitude of the rescored value in range.

## 5 Refinement

The refinement phase considers the overall strength of occurrence of each morpheme in the vocabulary. Thus, if a certain root morpheme is a true morpheme then all the pattern morphemes it occurs with would have higher scores since they also would be true morphemes. In such a case, this phase would increase the overall average strength for the root. The scores

obtained from the frequency-based method (Section 4) are frequency counts or weighted frequency counts. The scoring and rescoring in this refinement step differs in that it evaluates each root by averaging over scores of the corresponding patterns it occurs with in the dataset. We again iteratively refine based on the previous scores for  $k=0,1,2,\dots,m$  iterations,

$$S_j(r_t^x) = \frac{1}{f_r} \sum_{i=1}^N \left( S_{k-1}(p_{w_i}) \mid r_t^x = r_{w_i} \right) \quad (\text{Eq. 7})$$

where  $f_r$  is the number of words with root  $r$ , from a total of  $N$  vocabulary words. Similarly, for the pattern rescoring with the best so far pattern,  $p_w^b$ ,

$$S_j(p_w^x) = \frac{1}{f_p} \sum_{i=1}^N \left( S_{k-1}(r_{w_i}) \mid p_w^x = p_{w_i} \right) \quad (\text{Eq. 8})$$

Here we sum over the score of counterpart morphemes based on the match of the target morpheme in a vocabulary word, unlike the rescoring step, where we match the corresponding roots.

## 6 Morphological Analysis

A word,  $w_i$ , is analysed into its potential root and pattern template by considering every possible combination of trilateral root and corresponding pattern pairs,  $\langle r^x, p^x \rangle$ , as defined in equation (Eq. 1). Each analysis is scored with the sum of the scores for the root,  $r^x$ , and pattern,  $p^x$ , in the root lexicon and pattern lexicon, respectively. While combining scores we again apply scaling as in equation (Eq. 4) in order to guarantee equal contributions from each morpheme. The analysis,  $x$ , with the highest score, as calculated in equation (Eq. 9), is selected and is output.

$$\max_{x=1..n} \left( S(r_w^x) + SS(p_w^x) \right) \quad (\text{Eq. 9})$$

Since we are considering text without diacritics, due to the absence of short vowels, we only expect words to contain single letter infixes. Hence we also experiment with an alternative configuration of the word decomposition,  $\langle r^z, p^z \rangle$  in which only those tuples with single character infixes in patterns are considered for analysis, and all other tuples are dropped. We refer to this configuration as 'IF1' in the following evaluation.

## 7 Evaluation

The evaluation dataset comes from the Quranic Arabic Corpus (QAC),<sup>1</sup> which contains approximately 7370 undiacritized, stemmed token types. Although for evaluation purposes we use the stemmed vocabulary provided by QAC, such stemmed words could be obtained using existing techniques for unsupervised concatenative morphology learning (e.g. Poon *et al.*, 2009).

More than 7192 words (95% of the total vocabulary) are tagged with their root forms since the Quran consists mostly of words of derivable forms, with very few proper nouns. Sometimes alterations in root radicals take place, for example, in hollow roots, when moving from a root containing a long vowel to the surface word, the long vowel might change its form to another type or get dropped. Such words with hollow roots or reduplicated radicals, whose characters do not match every radical of the root, were excluded from the evaluation as they are beyond the scope of the learning algorithm. After these exclusions, 5468 word and root evaluation pairs remain.

### 7.1 Root Identification

We evaluate morphological analysis through correct identification of the root. Accuracy is measured in terms of the percentage of roots that are correctly identified.

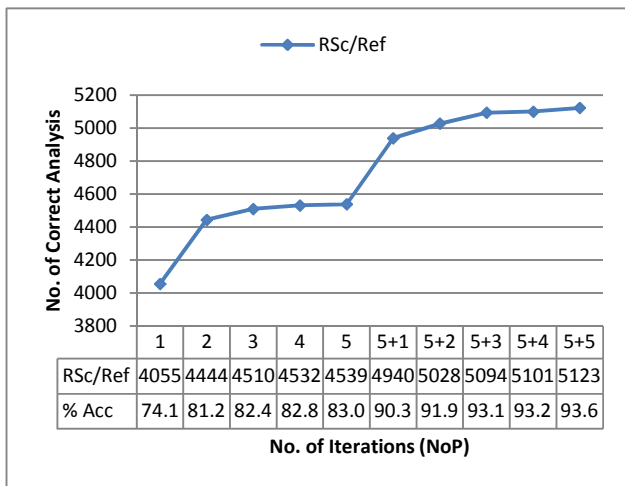


Figure 2. Results for iterative scoring and refinement.

Using the initial frequency-based scoring function (Section 4.1), we obtain a baseline (BL) accuracy of 74.1%. Figure 2 shows the results of

the rescoring (RSc) and refinement (Ref) phases, with  $n=5$  and  $m=5$  iterations respectively (NoP  $n+m$ ). There is a sudden improvement in accuracy after the first rescoring phase, and gradual improvement thereafter until the fifth. The refinement phase shows a similar trend, with a sudden improvement in accuracy at NoP 5+1. Here too the improvement is more gradual after each further iteration.

Configuration	Total Correct	Percentage Correct (%)
Baseline (BL)	4055	74.2
RSc_NoP1	4444	81.2
RSc_NoP5	4539	83.0
RSc_Ref_NoP5+1	4940	90.3
RSc_Ref_NoP5+5	5123	93.6
RSc_Ref_IF1	5159	<b>94.3</b>

Table 1. Results at key stages.

Table 1 shows the number of correct results at key stages. Rescoring and refinement each improve accuracy by 7 percentage points on their first iteration. This shows the advantage of using weighted morpheme scores. The subsequent iterations give total improvements of approximately 3 points. The IF1 configuration yields a further improvement of 0.75 points, indicating that some irrelevant analyses have been filtered out. With all the enhancements, the overall accuracy of 94.3% is an improvement of more than 20 percentage points over the baseline.

## 8 Conclusions and Future Directions

We have presented a novel, unsupervised approach to learning non-concatenative morphology. The approach learns trilateral roots and pattern templates, based on the idea that each may be revealed by their converses, using a mutually recursive scoring method. A subsequent refinement phase further increases accuracy.

The approach could be extended to roots beyond trilateral by adapting the scoring function to accommodate for morpheme length. In the future, we intend to apply the method to learning other kinds of morphological structures.

### Acknowledgments

We thank the referees for valuable comments, and in particular pointing out the correspondence to the hubs and authorities algorithm.

<sup>1</sup> <http://corpus.quran.com/>

## References

- Mohamed Attia Ahmed, 2000. *A Large-Scale Computational Processor of the Arabic Morphology, and Applications*. Master's Thesis, Faculty of Engineering, Cairo University, Cairo, Egypt.
- Kenneth Beesley. 1996. Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Copenhagen, Denmark, 89-94.
- Alexander Clark. 2007. Supervised and unsupervised learning of Arabic morphology. *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 181-200.
- Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR '05)*, 106-113.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1-3):1-33.
- Kareem Darwish. 2002. Building a shallow morphological analyzer in one day. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA, 1-8.
- Guy De Pauw and Peter Wagacha. 2007. Bootstrapping morphological analysis of Gikuyu using unsupervised maximum entropy learning. In *Proceedings of the Eighth Annual Conference of the International Speech Communication Association*. Antwerp, Belgium.
- John Goldsmith. 2000. Linguistica: An automatic morphological analyser. In *Proceedings of the 36th Meeting of the Chicago Linguistic Society*. 125-139.
- John Goldsmith. 2006. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(4):353-371.
- Harold Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309-350.
- Jon Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604-632.
- Hoifung Poon, Colin Cherry and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of the Conference of the North American Chapter of the ACL*, Boulder, CO, 209-217.
- Paul Rodrigues and Damir Čavar. 2005. Learning Arabic morphology using information theory. In *Proceedings of the Chicago Linguistics Society. Vol 41*. Chicago: University of Chicago. 49-58.
- Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the Conference of the North American Chapter of the ACL*, Pittsburgh, PA, 183-191.
- Aris Xanthos. 2008. *Apprentissage Automatique de la Morphologie: Le Cas des Structures Racine-Schème 'The Automatic Learning of Morphology: The Case of Root-and-Pattern Structures'*. Berne, Switzerland: Peter Lang.

# Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text

Minh Quang Nhat Pham, Minh Le Nguyen and Akira Shimazu

Japan Advanced Institute of Science and Technology

1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN

{minhpn, nguyenml, shimazu}@jaist.ac.jp

## Abstract

Finding contradiction text is a fundamental problem in natural language understanding. Previous work on finding contradiction in text incorporate information derived from predicate-argument structures as features in supervised machine learning frameworks. In contrast to previous work, we combine shallow semantic representations derived from semantic role labeling with binary relations extracted from sentences in a rule-based framework. Evaluation experiments conducted on standard data sets indicated that our system achieves better recall and F1 score for contradiction detection than most of baseline methods, and the same recall as a state of the art supervised method for the task.

## 1 Introduction

Contradiction detection (CD) in text is a fundamental task in natural language understanding, and necessary for many applications (De Marneffe et al., 2008; Voorhees, 2008). For instance, contradictions need to be recognized by question answering systems or multi-document summarization systems (Harabagiu et al., 2006). The task is to detect whether the contradiction relationship exists in a pair of a text  $T$  and a hypothesis  $H$ .

There are several approaches to the CD task. Contradiction detection can be formalized as a binary classification problem (Harabagiu et al., 2006; De Marneffe et al., 2008). The main effort of work which adopt this approach is to find out effective features for recognizing contradiction. The other approach is using functional relations indicated by verb or noun phrases for detecting contradiction (Ritter et al., 2008).

Beyond string-based matching approaches, one can approach to the CD task by applying logical

inference techniques. Although the logical inference approach may obtain good precision, it is not widely used for the task due to the fact that full predicate-logic analysis is currently not practical for wide-coverage semantic processing (Burchardt et al., 2009). Given that fact, (Burchardt et al., 2009) pointed out that using shallow semantic representations based on predicate-argument structures and frame knowledge is an intuitive and straightforward approach to textual inference tasks.

In contrast to previous work which integrate predicate-argument structures as features in machine learning-based systems (Harabagiu et al., 2006; De Marneffe et al., 2008), this paper combines shallow semantic representations derived from semantic role labeling with binary relations extracted from sentences for the CD task. The proposed system consists of two modules. The first module relies on the alignment of semantic role (SRL) frames extracted from the text and the hypothesis in each pair while the second one performs contradiction detection over binary relations extracted from the pair. If the SRL-based module fails to identify the contradiction relationship in the pair, the second module will be applied. We expect that the second module will improve the coverage of the first one. Evaluation experiments on standard data sets obtained from RTE challenges (Giampiccolo et al., 2007; Giampiccolo et al., 2008; Bentivogli et al., 2009) show that the proposed system achieves better recall and F1 score for contradiction detection than most of baseline methods, and the same recall as a state of the art supervised method for the task.

## 2 Linguistic Analysis

After parsing the text and the hypothesis of a pair by using Stanford CoreNLP<sup>1</sup>, we utilize SENNA

<sup>1</sup>Stanford CoreNLP is available online on: <http://nlp.stanford.edu/software/corenlp.shtml>

package<sup>2</sup> (Collobert et al., 2011) for semantic role labeling. Then, we extract SRL frames from the output of SENNA. An SRL frame consists of a verb predicate and a list of SRL elements.

In the system, we use REVERB (Fader et al., 2011) – a tool which can automatically identify and extract binary relations from English sentences. The input of REVERB is a POS-tagged and NP-chunked sentence and its output is a set of extraction triples of the form  $(arg1, R, arg2)$ , in which  $R$  represents the relation phrase between two arguments:  $arg1$  and  $arg2$ .

REVERB cannot extract some useful relations such as “isA” relations which specify the equivalent relation of two objects. In addition, in some cases, relation phrases of two extraction triples cannot be compared without using inference rules that specify the entailment relationship between two triples. Therefore, we propose several simple heuristic methods to extract additional binary relations from a text segment.

First, we extract “isA” relations from three information sources: i) co-reference resolution information; ii) noun phrases which the ending parts are recognized as a named entity;; and iii) “abbrev” relations in dependency parses.

Second, entailment rules or inference rules which specify directional entailment relations between two text fragments have been shown to be useful for RTE and question answering (Berant et al., 2011). In this study, we transform triples generated by REVERB by looking up the corpus of 30,000 entailment rules between typed predicates obtained from (Berant et al., 2011).

### 3 Contradiction Detection by Matching Semantic Frames

Let us denote an SRL frame by a tuple  $S = \{V, E_1, \dots, E_k\}$ , where  $V$  is used to denote the verb predicate; and  $E_i$  represents the  $i$ -th SRL element in the frame. Each SRL element has a type and underlying words. Types of SRL elements follow the annotation guideline in PropBank (Palmer et al., 2005). SRL elements can be arguments or modifiers (adjuncts). We denote two sets of SRL frames of  $T$  and  $H$  by  $T = \{S_i^{(t)}\}_{i=1}^m$  and  $H = \{S_j^{(h)}\}_{j=1}^n$ , in which  $m$  and  $n$  are the number of SRL frames extracted from  $T$  and  $H$ , respectively.

<sup>2</sup>SENNA is available online on: <http://ml.nec-labs.com/senna/>

### 3.1 Contradiction Detection Model

The contradiction detection model consists of a contradiction function  $\mathcal{F}_S(T, H)$  which calculates the contradiction measurement for the pair  $(T, H)$  on their SRL frames. Then,  $\mathcal{F}_S(T, H)$  is compared with a threshold value  $t_1$ . If  $\mathcal{F}_S(T, H) \geq t_1$ , we determine that  $T$  and  $H$  are contradictory.

In order to define the contradiction function  $\mathcal{F}_S(T, H)$ , we rely on the assumption that  $T$  and  $H$  are contradictory if there exists an event indicated by an SRL frame in  $H$ , which is incompatible with an event indicated by  $T$ . Formally, the function  $\mathcal{F}_S(T, H)$  is defined as following:

$$\mathcal{F}_S(T, H) = \max_{S_i^{(t)} \in T, S_j^{(h)} \in H} \mathbf{f}(S_i^{(t)}, S_j^{(h)}), \quad (1)$$

where  $S_i^{(t)}$  and  $S_j^{(h)}$  are two SRL frames in  $T$  and  $H$ , respectively; and  $\mathbf{f}(S_i^{(t)}, S_j^{(h)})$  is a contradiction function defined on the two SRL frames.

Next, we define the function  $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$  of two SRL frames  $S_1^{(t)} \in T$  and  $S_2^{(h)} \in H$ . For concreteness, we denote  $S_1^{(t)} = \{V_1, E_1^{(1)}, \dots, E_k^{(1)}\}$  and  $S_2^{(h)} = \{V_2, E_1^{(2)}, \dots, E_\ell^{(2)}\}$ .

The function  $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$  relies on the alignment of SRL elements across two frames. Since the number of SRL elements in an SRL frame is not very large, we propose a greedy alignment algorithm that considers all possible pairs of an SRL element in  $S_1^{(t)}$  and an SRL element in  $S_2^{(h)}$ . The core part of the greedy algorithm is the similarity measure between two SRL elements. We apply the local lexical level matching method (Dagan et al., 2007) to calculate the similarity of two SRL elements. In addition, we utilize co-reference resolution information by substituting mentions found in an SRL element with their equivalent mentions in the corresponding co-reference chain.

After generating the alignment between elements of two SRL frames, we define the contradiction function  $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$  as follows.

From the rationale that two events are not contradictory if they are not related, we filter out “not contradictory” SRL frame pairs by calculating their relatedness. The relatedness of two SRL frames is defined as product of the relatedness of their verb predicates and SRL elements:

$$R(S_1^{(t)}, S_2^{(h)}) = R(V_1, V_2) \times \max_{i,j} R(E_i^{(1)}, E_j^{(2)}), \quad (2)$$

where  $R$  represents the relatedness between two items;  $E_i^{(1)} \in S_1^{(t)}$  and  $E_j^{(2)} \in S_2^{(h)}$  are SRL elements;  $V_1$  and  $V_2$  are verbs of  $S_1^{(t)}$  and  $S_2^{(h)}$ , respectively.

The relatedness of two verbs is assigned to 1.0 if their relation is found in WordNet (Fellbaum, 1998) or in VerbOcean database (Chklovski and Pantel, 2004). In other cases, we employ WordNet::Similarity package (Pedersen et al., 2004) to compute the similarity of two verbs. The relatedness of two SRL elements  $E_i^{(1)}$  and  $E_j^{(2)}$  is defined as the local lexical level matching score.

The relatedness of two SRL frames is compared with a threshold. If it is below the threshold, then  $S_1^{(t)}$  and  $S_2^{(h)}$  are not related.

If two SRL frames are related, we consider two situations: 1) two verb predicates are matching and 2) Two verb predicates are opposite. Note that if two verb predicates are neither matching nor opposite,  $f(S_1^{(t)}, S_2^{(h)})$  is also assigned to 0.

In the system, that two verbs are matching are determined by utilizing synonyms in WordNet and WordNet-base semantic similarity. If two verb are matching, the function  $f(S_1^{(t)}, S_2^{(h)})$  is defined based on the alignment generated in the alignment process. We use the incompatibility of aligned arguments and modifiers such as temporal, location, or negation modifiers to calculate  $f(S_1^{(t)}, S_2^{(h)})$ .

In the second case, two verbs are opposite if they are found as antonym verbs in WordNet or opposite verbs in VerbOcean. In this case, the contradiction function  $f(S_1^{(t)}, S_2^{(h)})$  is defined as the similarity of their SRL elements. We define the element-based similarity of two frames as the product of similarity scores of the aligned elements having the same type.

#### 4 Contradiction Detection by Relation Matching

The main idea of this module is as follows. In the first step, we extract triples from  $T$  and  $H$  by using REVERB tool and our heuristics. Next, we compare each triple in  $H$  with every triple in  $T$ , and determine whether the contradiction relationship exists in some pairs of triples.

Formally, we denote a extraction triple by  $(x, r, y)$  where  $x$  and  $y$  respectively represent the first and second argument, and  $r$  represents the relation phrase of the triple.

We denote  $T = \{(x_i^{(t)}, r_i^{(t)}, y_i^{(t)})\}_{i=1}^m$  and  $H =$

$\{(x_j^{(h)}, r_j^{(h)}, y_j^{(h)})\}_{j=1}^n$ . Here,  $m$  and  $n$  are respectively the numbers of triples in  $T$  and  $H$ . The contradiction detection task is reduced to searching for incompatible triple pairs across  $T$  and  $H$ . We define the contradiction function on triples of  $T$  and  $H$  as follows.

$$\mathcal{F}_T(T, H) = \max_{T_i \in T; H_j \in H} \mathbf{g}(T_i, H_j), \quad (3)$$

where  $T_i$  is the  $i$ -th triple of  $T$ ;  $H_j$  is the  $j$ -th triple of  $H$ ; and  $\mathbf{g}(T_i, H_j)$  is the contradiction function of the two triples  $T_i$  and  $H_j$ .

The function  $\mathbf{g}(T_i, H_j)$  is based on the mismatch of two triples  $T_i$  and  $H_j$ . We consider three cases as follows. If their relation phrases and first arguments are matching, the mismatch of second arguments will be calculated. If two relation phrases are matching and roles of arguments in the two triples are exchanged,  $\mathbf{g}(T_i, H_j)$  is assigned to 1.0. However, this rule is not applied for “isA” (equivalent) relations. In contrast, if two relation phrases are opposite, the similarity measures of first arguments and second arguments are taken into account.

In the procedure for calculating  $\mathbf{g}(T_i, H_j)$ , we need to determine whether two relation phrases  $r_i^{(t)}$  and  $r_j^{(h)}$  are matching or not. If the surface and base forms of two relation phrases are different, we use WordNet to detect whether main verbs of  $r_i^{(t)}$  and  $r_j^{(h)}$  are synonyms. In order to check if two relation phrases  $r_i^{(t)}$  and  $r_j^{(h)}$  are opposite or not, we utilize antonym relations in WordNet and opposite relations in VerbOcean.

In the module, that two arguments are matching is checked by using their similarity. The similarity score of two arguments is computed by the same method as that for computing the similarity of two SRL elements. When we detect the contradiction of two arguments, we use the contradiction rule as follows. Two arguments are contradictory if they include two entities having the same type but different values. Especially, we take into account four categories: NUMBER, DATE, TIME, and LOCATION. In other cases, we use the similarity of two arguments as the evidence for contradiction detection.

## 5 Evaluation Experiments

### 5.1 Data Sets

In experiments, we evaluate the proposed method on the test sets of the three-way subtask at RTE-



Table 1: Label distribution in three test sets

Data Set	Contradiction	Entailment	Unknown	Total
RTE-3 Test	72	410	318	800
RTE-4 Test	150	500	350	1000
RTE-5 Test	90	300	210	600

3, RTE-4, and RTE-5 competitions (Giampiccolo et al., 2007; Giampiccolo et al., 2008; Bentivogli et al., 2009). The development sets provided at each competition are used to tuned threshold values in two CD modules of the system. The three-way subtask requires participant systems to decide whether the entailment, contradiction, or unknown relationship exists in a pair. Since in this study, we focus on contradiction relationship in a text pair, entailment and unknown labels in data sets are converted into non-contradiction labels. Table 1 provides statistics on the test sets of three-way subtask in RTE-3, RTE-4, and RTE-5.

The data sets used in experiments are unbalanced, so the average accuracy over all labels is not an appropriate evaluation measures. Therefore, we use Precision, Recall, and F1 scores of the contradiction label as evaluation measures.

## 5.2 Baseline Methods

The first baseline method is the method presented in (De Marneffe et al., 2008), which employed supervised machine learning techniques for the CD task. To the best of our knowledge, (De Marneffe et al., 2008) is the only contradiction detection-focused work that evaluates on data sets of RTE challenges.

The second baseline is the BLUE system of Boeing’s team (Clark and Harrison, 2009) at RTE-4 and RTE-5 competitions. The BLUE system adopted the logical inference approach to RTE, which performs inference on logic-based representations of the text and the hypothesis in a pair. We use best scores among submitted runs of the BLUE system at each competition.

In experiments, we also compare the results achieved by our system with average results of submitted systems for three-way subtask at RTE-3, RTE-4 and RTE-5 challenges. The numbers submitted systems in RTE-3, RTE-4 and RTE-5 for the three-way subtask are 12, 34, and 24 submissions, respectively.

In order to assess the effectiveness of the two-stage system scheme, we separately run each CD module on the three data sets and compare the re-

sults with those of the combined system.

## 5.3 Experimental Results

Table 2 provides experimental results achieved on test sets of RTE-3, RTE-4, and RTE-5 challenges by our system and baseline methods. As shown in results, the proposed system consistently obtained better recall values and F1 scores than those of baseline methods except the supervised machine learning-based method in (De Marneffe et al., 2008). Compared with the method presented (De Marneffe et al., 2008), our system achieves the same recall but lower precision.

The results shown in Table 2 indicated that the SRL-based module consistently achieved better recall and F1 score than those of the triple-based module. A possible explanation is that the information contained in shallow semantic representations is richer than that of extraction triples, so the SRL-based module covers more contradiction phenomena than the triple-based module. As expected, the combined system consistently obtained better recall and F1 score than each separate module. Experimental results confirmed our observation that the second backup module increases the coverage of contradiction phenomena for our system.

## 6 Conclusion

In this paper, we have presented a new rule-based method for finding contradiction in text, which combines shallow semantic representations with binary relations extracted from sentences. We define contradiction measurements on the predicate-argument structures and binary relations extracted from the text and the hypothesis in a pair. We deal with the low-coverage problem of semantic role resources by using a backup module which exploits extraction triples. Experimental results achieved on standard data sets showed that our proposed system obtained better recall and F1 score for contradiction detection than most of baseline methods.

Table 2: Experimental results on three data sets

Method	RTE-3 Pilot			RTE-4 Test			RTE-5 Test		
	P	R	F1	P	R	F1	P	R	F1
De Marneffe (2008)	22.95	19.44	21.04	–	–	–	–	–	–
BLUE system	–	–	–	41.67	10.0	16.13	42.86	6.67	11.54
Average result	10.72	11.69	11.18	25.26	13.47	13.63	26.40	13.70	14.79
SRL-based	13.41	15.27	14.28	22.41	17.33	19.55	22.72	16.67	19.23
Triple-based	22.58	9.72	13.59	26.3	10.0	14.49	19.48	16.67	17.96
<b>Two-stage (our system)</b>	<b>14.0</b>	<b>19.44</b>	<b>16.27</b>	<b>23.0</b>	<b>22.67</b>	<b>22.82</b>	<b>21.14</b>	<b>28.89</b>	<b>24.4</b>

## References

- L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proceedings of TAC Workshop*.
- Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Aljoscha Burchardt, Marco Pennacchiotti, Stefan Thater, and Manfred Pinkal. 2009. Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering*, 15(Special Issue 04):527–550.
- Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.
- Peter Clark and Phil Harrison. 2009. Recognizing textual entailment with logical inference. In *In Proceedings of the First Text Analysis Conference (TAC 2008)*.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 999888:2493–2537, November.
- Ido Dagan, Dan Roth, and Fabio Massimo. 2007. A tutorial on textual entailment.
- Marie-catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. In *In Proceedings of ACL 2008*.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- 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.
- Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, Elena Cabrio, and Bill Dolan. 2008. The fourth pascal recognizing textual entailment challenge. In *In Proceedings of TAC 2008 Workshop*.
- Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast, and contradiction in text processing. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, March.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004, HLT-NAACL–Demonstrations '04*, pages 38–41, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Alan Ritter, Stephen Soderland, Doug Downey, and Oren Etzioni. 2008. It’s a contradiction – no, it’s not: A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 11–20, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Ellen M. Voorhees. 2008. Contradictions and justifications: Extensions to the textual entailment task. In *Proceedings of ACL-08: HLT*, pages 63–71, Columbus, Ohio, June. Association for Computational Linguistics.

# Using Transliteration of Proper Names from Arabic to Latin Script to Improve English-Arabic Word Alignment

**Nasredine Semmar**

Institut CEA LIST, Laboratoire Vision  
et Ingénierie des Contenus  
CEA Saclay – Nano-INNOV, F-91191  
Gif-sur-Yvette Cedex, France  
nasredine.semmar@cea.fr

**Houda Saadane**

LIDILEM, Université Grenoble III  
Domaine Universitaire  
1180, avenue centrale, F-38400 Saint  
Martin d'Hères, France  
houda.saadane@e.u-  
grenoble3.fr

## Abstract

Bilingual lexicons of proper names play a vital role in machine translation and cross-language information retrieval. Word alignment approaches are generally used to construct bilingual lexicons automatically from parallel corpora. Aligning proper names is a task particularly difficult when the source and target languages of the parallel corpus do not share a same written script. We present in this paper a system to transliterate automatically proper names from Arabic to Latin script, and a tool to align single and compound words from English-Arabic parallel texts. We particularly focus on the impact of using transliteration to improve the performance of the word alignment tool. We have evaluated the word alignment tool integrating transliteration of proper names from Arabic to Latin script using two methods: A manual evaluation of the alignment quality and an evaluation of the impact of this alignment on the translation quality by using the open source statistical machine translation system Moses. Experiments show that integrating transliteration of proper names into the alignment process improves the F-measure of word alignment from 72% to 81% and the translation BLEU score from 20.15% to 20.63%.

## 1 Introduction

Bilingual lexicons of proper names play a vital role in Machine Translation (MT) and Cross-Language Information Retrieval (CLIR). Word alignment approaches are generally used to construct bilingual lexicons automatically from parallel corpora. Aligning proper names requires both recognition of the proper names present in

the parallel corpus and their alignment (Abuleil and Evens, 2004). This task is particularly difficult when the source and target languages of the parallel corpus do not share a same written script. A solution to this issue consists in writing the proper names present in the parallel corpus in the same written script. This operation is named transliteration and consists in replacing each grapheme of a writing system by another grapheme or a group of graphemes of another writing system, regardless of pronunciation.

In order to study the impact of using transliteration to improve the performance of a word alignment tool, we present in this paper a system to transliterate automatically proper names from Arabic to Latin script, and a tool to align single and compound words from English-Arabic parallel texts.

The remainder of the paper is organized as follows: Section 2 recalls in some previous work addressing tasks of transliteration and bilingual lexicon extraction from parallel corpora. In section 3, we present briefly the system for automatic transliteration of proper names from Arabic to Latin script. Section 4 describes the process of using transliteration in the word alignment tool. We present in section 5 the experimental protocol we followed and discuss the obtained results. We finally conclude and present directions for future work in section 6.

## 2 Related Work

In order to build bilingual lexicons from parallel corpora automatically, several word alignment approaches have been explored (Daille *et al.*, 1994; Blank, 2000; Barbu, 2004). These approaches align proper names correctly when the source and target languages of the parallel corpus

share a same written script. Recent research works for aligning proper names when the source and target languages do not share a same written script have focused on automatic alignment of transliterations in order to enrich bilingual lexicons of named entities. These include (Al-Onaizan and Knight, 2002) and (Sherif and Kondrak, 2007) who worked on the Arabic-English alignment, (Tao *et al.*, 2006) who worked on Arabic, Chinese and English, and (Shao and Ng, 2004) who used the information resulted from transliterations based on pronunciation. They combined the obtained information from the translation context and those generated from Chinese and English transliteration. This technique allowed processing some specific infrequent words. Some other systems assign for a given name only one transliteration such as the generative model for English words written in Japanese (Katakana) to Latin transcription (Knight and Graehl, 1997). This approach was adapted by (Stalls and Knight, 1998) to translate English words written in Arabic into English. (AbdulJaleel and Larkey, 2003) proposed a system based on a statistical approach to transliterate English names into Arabic. This system has several limitations as it uses the computation of the most probable form supposed to be the correct one. Indeed, this hypothesis is not always valid in all the Arab countries and dialects. To avoid pronunciation and dialect varieties, (Alghamdi, 2005) proposed a system to transliterate vowelized Arabic names into English. This system is based on a dictionary of Arabic names in which the pronunciation is set using vowels added to listed names with an indication of their equivalents in English. This approach has a strong limitation when used in word alignment as it proposes only one transliteration for a given name. Recently, (Saadane *et al.*, 2012) proposed an approach to transliterate proper names from Arabic to Latin script which takes into account phonological and linguistic aspects. The authors reported an improvement of the F-measure of their French-Arabic word alignment tool from 82% to 86%.

### 3 Transliteration of Proper Names from Arabic to Latin Script

The transliteration system of proper names from Arabic to Latin script used in this study (Saadane *et al.*, 2012) is based on a finite-state automaton. This automaton switches from one state to another according to the outward transitions of the

current state and the currently processed letter of the Arabic word. The transliteration process is composed of the following main steps:

1. **Transliteration:** Each proper name is, first, split or not into several elements according to its type and the particles which do not compose the name itself are transcribed. Then, transliteration rules are applied to transliterate the names themselves. These rules are applied in a certain order based on the number of consonants of the proper name. For example, the compound name “عبد الرشيد” is, first, split into “عبد + ال + رشيد”, second, the particles “عبد” and “ال” are transcribed into “abd” and “al”, and finally the name “رشيد” is transliterated into rachid, rashid, etc.
2. **Normalization:** This step consists in performing some post-processing on the generated transliterations such as changing the first letter into capital.
3. **Weighting:** This step consists in assigning weights to the rules used to generate the list of transliterations in order to display the results sorted from the most likely to the least likely. Results of some search engines are exploited to compute these weights based on the number of occurrences for each generated transliteration of the proper name.

### 4 Alignment of Proper Names from English-Arabic Corpora

Word alignment from parallel corpora consists, on the one hand, in identifying words present in the source and target texts, and, on the other hand, in establishing correspondences between these words. The word alignment tool evaluated in this study (Semmar *et al.*, 2010), first, identifies single words and compound words present in the parallel corpus using the linguistic analyzer LIMA (Besançon *et al.*, 2010), and, second, establishes correspondence relations between these words using the following steps:

1. Look-up of words which are present in an existing English-Arabic lexicon composed of 149495 entries;
2. Matching of words which are cognates;
3. Matching of words which have the same grammatical categories;
4. Establishing correspondence relations between compound words.

We describe below only the step 2 which illustrates the process of using transliteration of proper names from Arabic to Latin script in English-Arabic word alignment.

Proper names alignment consists, first, in searching words present in the source and target sentences which have the grammatical category “Proper Name” by using the results of the linguistic analyzer LIMA, and, second, in identifying words which are cognates. Several research works have shown that using cognates can improve both sentence alignment (Simard *et al.*, 1993) and word alignment (Kondrak, 2005). In our implementation, we consider, in a first step, that pairs of words which share the first four characters as cognates. This step uses the results of the transliteration into Latin script of all the proper names present in the Arabic corpus and can identify, for example, that the proper name “Kosovo” and the transliteration of the Arabic word “كوسوفو” (“kosoufou”) are cognates. However, this step does not detect pairs of words such as “Algeria” and “aljazair” (transliteration of the Arabic word “الجزائر”). To take into account this kind of pairs of words, we used the Jaro–Winkler distance  $DJW$  (Winkler, 1990), a similarity measure based on the number of letters in common between the string of the word of the source language  $ws$  and the string of the word of the target language  $wt$ .

$$DJ(ws, wt) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left( \frac{m}{|ws|} + \frac{m}{|wt|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases}$$

where:

- $m$  is the number of matching characters. Two characters from  $ws$  and  $wt$  respectively, are considered matching only if they are the same and not farther than:

$$\left( \frac{\max(|ws|, |wt|)}{2} \right) - 1$$

- $t$  is the number of transpositions which is equal to the half of number of characters in  $ws$  that do not line up (by index in the matched subsequence) with identical characters in  $wt$ .
- $|ws|$ ,  $|wt|$  are lengths of the strings corresponding to the words  $ws$  and  $wt$ .

Jaro–Winkler similarity measure is a variant of the Jaro distance metric  $DJ$  (Jaro, 1989).

$$DJW(ws, wt) = DJ(ws, wt) + (lp(1 - DJ(ws, wt)))$$

where:

- $l$  is the length of common prefix at the start of the string up to a maximum of 4 characters.
- $p$  is a constant scaling factor for how much the score is adjusted upwards for having common prefixes.

In order to identify the values of  $l$  and  $p$  which provide the best alignment, we checked manually the result of the transliteration of 254 proper names. This evaluation showed that, if  $l$  is equal to 2 and  $p$  is equal to 0.1, the words  $ws$  and  $wt$  are cognates when the value of the Jaro–Winkler distance is the highest. Table 1 presents results after running our word alignment tool on the English sentence “Condemning all violations of human rights in Kosovo which have affected all ethnic groups in Kosovo.” and its Arabic translation “وإذ تدن كل ما ارتكب في كوسوفو من انتهاكات لحقوق الإنسان طالت جميع الفئات العرقية في كوسوفو.”

Lemmas of words of the source sentence	Lemmas of words of the target sentence
condemn	أَذَانَ
violation	إِنْتِهَاكَ
human	إِنْسَانَ
right	حَقَّ
Kosovo	كوسوفو
affect	طَالَ
ethnic	عَرَقِيَّةَ
group	فِئَةَ
Kosovo	كوسوفو
violation_human_right	إِنْتِهَاكَ حَقَّ إِنْسَانَ
human_right	حَقَّ إِنْسَانَ
ethnic_group	فِئَةَ عَرَقِيَّةَ

Table 1. Results of single words and compound words alignment

The word “Kosovo” was aligned using cognates matching after transliteration, the words “condemn”, “human”, “affect” and “group” were aligned using grammatical categories matching and the other single words exist in the English-Arabic lexicon. The compound words “violation\_human\_right”, “إِنْتِهَاكَ حَقَّ إِنْسَانَ”, “human\_right”, “حَقَّ إِنْسَانَ”, “ethnic\_group” and “فِئَةَ عَرَقِيَّةَ” are first recognized by LIMA respectively from the source sentence and the target sentence, and then aligned using lexical and syntactic transfer rules between source and target languages (Ozdowska, 2004).

## 5 Experimental Results and Evaluation

The impact of using transliteration of proper names on the quality of alignment and machine translation has been evaluated according to the two following approaches:

- A manual evaluation comparing the results of our word aligner with a reference alignment;
- An automatic evaluation by integrating the results of our word aligner tool in the training corpus used to build the translation table of the statistical MT system Moses (Koehn *et al.*, 2007).

In order to evaluate the alignment quality manually, we used 500 English-Arabic aligned sentences extracted from the MT evaluation MEDAR<sup>1</sup> package and we followed the evaluation framework defined in (Mihalcea and Pedersen, 2003). Table 2 summarizes the results of our word aligner in terms of precision and recall. The first line describes the performance of the word aligner when it does not integrate transliteration and the second line mentions its performance when it uses transliteration. As we can see, the results demonstrate that using transliteration improves both precision and recall of word alignment. These results confirm those obtained by (Sajjad *et al.*, 2003) related to the improvement of alignment quality when integrating transliteration into the GIZA++ word aligner.

Alignment	Precision	Recall	F-measure
without using transliteration	0.90	0.60	0.72
with the use of transliteration	0.91	0.73	0.81

Table 2. Results of the evaluation of single and compound words alignment

The unavailability of a reference alignment of a significant size for single and compound words does not allow us to compare our approach with the state-of-the-art work. That's why we decided to study the impact of the use of transliteration in word alignment by integrating the results of our word aligner in the training corpus used to extract the translation model of Moses. The initial training corpus is composed of 75000 pairs of English-Arabic sentences extracted from the

<sup>1</sup> The MT evaluation MEDAR package is available on <http://www.medar.info/index.php>.

MEDAR corpus (2631654 English words and 2344878 Arabic words). We added to this corpus around 28000 pairs of single and compound words corresponding to the results of our word aligner which integrates transliteration applied on 1000 pairs of English-Arabic sentences. We also specified a language model for the target language using a corpus composed of 100000 Arabic sentences (3155516 words). The performance of the statistical machine translation system Moses is evaluated using the BLEU score on a test corpus composed of 500 pairs of sentences. Note that we consider one reference per sentence. The obtained results show that the inclusion in the training corpus of word alignment results integrating transliteration has improved the translation BLEU score from 20.15 to 20.63 (a gain of 0.48 points).

In order to assess statistical significance of the obtained results, we use the paired bootstrap resampling method (Koehn, 2004) which estimates the probability (*p-value*) that a measured difference in BLEU scores arose by chance by repeatedly (10 times) creating new virtual test sets by drawing sentences with replacement from a given collection of translated sentences. We carry out experiments using this method to compare the translation results without using transliteration and with the use of transliteration. At a 95% confidence interval (CI), the results vary from insignificant (at  $p > 0.05$ ) to highly significant. The *p-value* obtained is equal to 0.02 and therefore the improvement achieved by using transliteration is statistically significant.

## 6 Conclusion

We presented briefly in this paper a system to transliterate proper names from Arabic to Latin script and we proposed a tool to automatically align word pairs from an English-Arabic parallel corpus. We integrated the transliterated proper names into the cognates matching step and we obtained a gain of 9% on word alignment F-measure and a gain of 0.48 points in translation BLEU score. These encouraging results can be improved in a number of ways. First, we plan to affect a weight for each word pair in order to filter the word alignment results and to integrate them directly in the translation table of Moses. We also plan to use, on the one hand, the linguistic analyzer LIMA to lemmatize texts of the bilingual corpus, and on the other hand, factored models and a flexor to generate adequate surface forms from lemmas.

## References

- Saleem Abuleil and Martha Evens. 2004. *Named Entity Recognition and Classification for Text in Arabic*. The 13<sup>th</sup> International Conference on Intelligent & Adaptive Systems and Software Engineering, Nice, France.
- Nasreen AbdulJaleel and Leah S. Larkey. 2003. *Statistical transliteration for English-Arabic Cross Language Information Retrieval*. The 12<sup>th</sup> ACM International Conference on Information and Knowledge Management, New Orleans, LA, USA.
- Mansour Alghamdi. 2005. *Algorithms for Romanizing Arabic names*. Journal of King Saud University. Computer Sciences and Information. Riyadh, 17: Pages 1-27.
- Yaser Al-Onaizan and Kevin Knight. 2002. *Translating named entities using monolingual and bilingual resources*. The 40<sup>th</sup> ACL Conference, Philadelphia, USA.
- Ana M. Barbu. 2004. *Simple linguistic methods for improving a word alignment algorithm*. The 7<sup>th</sup> International Conference on the Statistical Analysis of Textual Data (JADT), Louvain, Belgium.
- Romarc Besançon, Gaël De Chalendar, Olivier Ferret, Faïza Gara, Meriama Laib, Olivier Mesnard, and Nasredine Semmar. 2010. *LIMA: A multilingual framework for linguistic analysis and linguistic resources development and evaluation*. The 7<sup>th</sup> international conference on Language Resources and Evaluation, Valletta, Malta.
- Ingeborg Blank. *Terminology extraction from parallel technical texts*. Véronis J. (Ed.), Parallel Text Processing, Dordrecht: Kluwer, 2000.
- Béatrice Daille, Eric Gaussier, and Jean. M. Langé. 1994. *Towards automatic extraction of monolingual and bilingual terminology*. The 15<sup>th</sup> International Conference on Computational Linguistics.
- Matthew A. Jaro. 1989. *Advances in record linkage methodology as applied to the 1985 census of Tampa Florida*. Journal of the American Statistical Association 84: Pages 414-420.
- Kevin Knight and Jonathan Graehl. 1997. *Machine transliteration*. Journal Computational Linguistics, 24(4): Pages 599-612.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicolas 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*. The Conference ACL 2007, demo session, Prague, Czech Republic.
- Philipp Koehn. 2004. *Statistical significance tests for machine translation evaluation*. The 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.
- Grzegorz Kondrak. 2005. *Cognates and Word Alignment in Bitexts*. The Tenth Machine Translation Summit (MT Summit X), Phuket, Thailand.
- Rada Mihalcea and Ted Pedersen. 2003. *An evaluation exercise for word alignment*. The Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, Edmonton, Canada.
- Sylwia Ozdowska. 2004. *Identifying correspondences between words: an approach based on a bilingual syntactic analysis of French/English parallel corpora*. The 20<sup>th</sup> International Conference on Computational Linguistics, Geneva, Switzerland.
- Houda Saadane, Nasredine Semmar, Ouafa Benterki and Christian Fluhr. 2012. *Using Arabic Transliteration to Improve Word Alignment from French-Arabic Parallel Corpora*. The fourth Workshop on Computational Approaches to Arabic Script-based Languages, AMTA 2012, San Diego, CA, USA.
- Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2011. *An algorithm for unsupervised transliteration mining with an application to word alignment*. The 49<sup>th</sup> Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, Pages 430-439, Portland, Oregon, USA
- Li Shao and Hwee Tou Ng. 2004. *Mining new word translations from comparable corpora*. The 20<sup>th</sup> International Conference on Computational Linguistics (COLING), Geneva, Switzerland.
- Tarek Sherif and Grzegorz Kondrak. 2007. *Bootstrapping a stochastic transducer for Arabic-English transliteration extraction*. The 45<sup>th</sup> ACL Conference, Prague, Czech Republic.
- Nasredine Semmar, Christophe Servan, Gaël de Chalendar, Benoît Le Ny, and Jean-Jacques Bouzaglou. 2010. *A Hybrid Word Alignment Approach to Improve Translation Lexicons with Compound Words and Idiomatic Expressions*. The 32<sup>nd</sup> Translating and the Computer Conference, England.
- Michel Simard, George F. Foster, and Pierre Isabelle. *Using cognates to align sentences in bilingual corpora*. 1993. The Conference of the Centre for Advanced Studies on Collaborative Research: Distributed computing, Volume 2, Pages 1071-1082.
- Bonnie Stalls and Kevin Knight. 1998. *Translating names and technical terms in Arabic text*. The COLING/ACL Workshop on Computational Approaches to Semitic Languages, Montreal, Canada.
- Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat, and ChengXiang Zhai. 2006. *Unsupervised named entity transliteration using temporal and phonetic correlation*. The 2006 EMNLP Conference, Pages 250-257.
- William E. Winkler. 1990. *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage*. Section on Survey Research Methods, American Statistical Association: Pages 354-359.

# A Semi-Supervised Method for Arabic Word Sense Disambiguation Using a Weighted Directed Graph

**Laroussi Merhbene**  
LATICE / Faculty of sciences of  
Monastir, Monastir, Tunisia  
Arous-  
si\_merhben@hotmail.co  
m

**Anis Zouaghi**  
LATICE, ISSAT Sousse,  
University of Sousse, Tu-  
nisia  
Anis.zouaghi@gmail  
.com

**Mounir Zrigui**  
LATICE, Faculty of sciences  
of Monastir, Monastir, Tun-  
isia  
mounir.zrigui  
@fsm.rnu.tn

## Abstract

In this paper, we propose a new semi-supervised approach for Arabic word sense disambiguation. Using the corpus and Arabic Wordnet<sup>1</sup>, we define a method to cluster the sentences containing ambiguous words. For each sense, we generate a cluster that we use to construct a semantic tree. Furthermore, we construct a weighted directed graph by matching the tree of the original sentence with semantic trees of each sense candidate. To find the correct sense, we use a similarity score based on three collocation measures that will be classified using a novel voting procedure. The proposed method gives a high rate of recall and precision.

## 1 Introduction

The human language is so complex to be learned. The syntactic form of words, the relation between a specific word form and its meaning are the basic parts of an intelligent system for the natural language processing.

In this work we aim to solve the task of identifying the sense of the ambiguous word. This task is called Word Sense Disambiguation (WSD), which is one of the oldest problems in natural language processing (NLP) (Agirre and Edmond, 2006). This work is part of a general framework of Arabic speech (Zouaghi, 2008).

In this work, we combine a supervised and an unsupervised method for Arabic word sense disambiguation. The innovative part in this work is

the construction of a semantic tree for each sense of the ambiguous word. Also we define a voting procedure that gives a weight for the score measure.

This paper is organized as follows. Section two describes the proposed method. The experimental results are described in section three. Finally this paper is concluded in section four.

## 2 Proposed method

We propose a semi-supervised method for Arabic word sense disambiguation.

For the unsupervised part of the proposed method, we use Arabic Wordnet (Black et al., 2006) and the corpus to construct sense clusters (group of sentences) characterizing a specific sense of the ambiguous word. Furthermore we construct a semantic tree for each sense of the ambiguous word.

The disambiguation procedure is based on the step of matching the semantic tree with the tree of the original sentence. We use a score measure (based on three collocation measures) to find the closest semantic tree to the tree of the sentence to be disambiguated.

The supervised part uses a voting procedure that will rank collocation measures during a classification task. The sense given by the measure having the highest rank will be attributed to the ambiguous word. In what follows we describe with more details each step cited above.

### 2.1 Construction of the sense clusters

In the first we apply some pre-treatment steps to glosses of the ambiguous word (definitions and synonyms extracted from Arabic wordnet) and sentences containing the ambiguous word (collected from the used corpus). Using the Kho-

---

<sup>1</sup> Arabic Wordnet is a concept dictionary with mappings between word definitions.



ja stemmer and the approximate string matching, we are able to construct the sense clusters. Some pre-treatment steps will be applied to these clusters. In what follows, we detail the steps of the proposed method.

**Pre-treatment:** Using the corpus we collect sentences containing the ambiguous word, we have to search the root of the ambiguous word (exp: for the word “العين” “alayn” we have to search the root “عين” “ayn”). The segmentation of these sentences is based on punctuation (., :, !, ?, etc.) and on the number of the words that have to be more than three.

Subsequently, we eliminate the stop-words that occur frequently in the corpus and they have no significant relation to the sense of the word. We use a general stop-list containing 29,985 stop-words, this list were elaborated by Arabic linguistics and judged as sufficient for the task of WSD.

**Root extraction:** We use the Khoja stemmer (Khoja,1999) for words contained in the glosses of the ambiguous word. Its advantage is that it uses a large linguistic data such as the list of verbal and noun patterns, stop-words, list of diacritic characters, etc.

For a specific word, this stemmer extracts the longest suffix and prefix, which will be matched with the existing list of patterns to extract the root. We notice that we use the list of stop-words in addition to the already used list (detailed in the previous paragraph).

**Sense Clustering:** The basic idea of Sense clustering is that the sentences representing the meaning of a particular sense are grouped in the same cluster  $C_i$  (Cluster of the  $i^{\text{th}}$  sense of the ambiguous word).

The list of sentences extracted from the corpus will be classified into clusters using the roots of the words containing in each gloss. To find the possible occurrences of the roots, we use the approximate string matching algorithm (Elloumi, 1998).

In the first we fill a matrix of the two words to be compared  $w_i$  and  $w_j$ . After that we use the step of back-tracking, to find the shortest common subsequence.

The words containing the common subsequence will be considered as occurrences of the stem. The Sentences containing the occurrences of stems obtained from glosses are grouped into

clusters representing each sense of the ambiguous word.

## 2.2 Semantic Tree construction

A text can be represented by Trees or graphs (co-occurrence graphs (Agirre and Sorora, 2007), collocation graphs (Klapaftis and Manandhar, 2008), semantic graphs (Plaza and Diaz, 2011)) that differs in the structure of text representation.

The first step is to transform the sentences of the clusters to binary trees,  $T = (N, E, R, RC, LC, L)$ , where:

- $N$  is a set of nodes,  $N = \{n_1 \dots n_2\}$ . Each node corresponds to a concept in the binary Tree.
- $E$  is a set of edges that represents the relation between the node  $n_i$  to the node  $n_j$ .
- $R$  is the root of the tree which is the ambiguous word.  $RC$  is the set of right children which are the words occurring on the right of the ambiguous word.
- $LC$  is the set of left children which are the words occurring on the left of the ambiguous word.
- $L$  is a function assigning the level of the nodes, it corresponds to their position regarding the ambiguous word.

Expect the root, each node of the tree has exactly one child. We denote  $\langle R, RC, LC \rangle$  a binary tree.

The second step is to merge all the obtained trees corresponding to the sentences contained in the same cluster. Accordingly, we obtain a semantic tree,  $ST = (N, E, R, C, L, Nb, H)$ , where:

- $C$  is the set of merged nodes,  $C = \{c_1, \dots c_n\}$ . The right and left child of each binary tree will be linked to the root of the semantic tree.
- $Nb$  is a function that returns the number of nodes in the semantic tree.
- $H$  is a function that returns the height of the semantic tree.

The step of merging trees uses an algorithm of breadth-first traversal, to find the repeated node that may have a higher level, same level or a lower level.

## 2.3 WSD procedure

In the first, we apply some pre-treatment steps to the original sentence containing the ambiguous word. The process of disambiguation is based on three steps:

### Step 1: Weighted directed graph construction:

We add edges weighted by the collocation measures between the nodes  $N_i$  of the tree of the original sentence (called  $T_{os}$ ) and the nodes  $N_j$  of the semantic tree of each sense (called  $ST_{S_k}$ , where  $S_k$  corresponds to the  $k^{th}$  sense).

This step called matching allows us to obtain a weighted directed graph. After eliminating stopwords, we extract the roots of the words contained in the original sentence. These roots are the nodes of the tree and the level in the tree  $T_{os}(N)$  will be affiliated corresponding to their position regarding the ambiguous word.

Each node of the tree extracted from the original sentence is matched with the nodes of the same level in the semantic tree of a particular sense. The links used for the matching step appear as a dashed line. They are weighted using one of the three collocation measures (Maning and Schütze, 1999) detailed in what follows:

#### The T-test

The T-test is measured as follows (see equation 1).

$$wc_{ij} = T = (\bar{x} - \mu) / \left( \sqrt{\frac{s^2}{N}} \right) \quad (1)$$

The mean of the distribution  $\mu$  is measured by multiplying  $P(w_i)$  to  $P(w_j)$ , where  $P(w)$  = number of occurrences of  $w$  in the corpus / Total number of words in the corpus.  $\bar{x}$  (sample mean) is equal to  $s^2$  (sample variance), measured by dividing the number of occurrences of the two words together by the total number of words in the corpus.

#### The Mutual Information

This measure determines how much a word can be informative for another word. The mutual information is measured as follows (see equation 2):

$$wc_{ij} = MI = \log_2 \frac{P(w_i, w_j)}{P(w_i) P(w_j)} \quad (2)$$

#### The Chi-Square $\chi^2$

The equation 3 in what follows details the measure of  $\chi^2$ .

$$wc_{ij} = \chi^2 = \frac{N \times (c_{1,1} \times c_{2,2} - c_{1,2} \times c_{2,1})^2}{(c_{1,1} + c_{1,2}) \times (c_{1,1} + c_{2,1}) \times (c_{1,2} + c_{2,2}) \times (c_{2,1} + c_{2,2})} \quad (3)$$

The basic principle is to count  $C_{1,1}$  (the number of occurrences of  $w_i$  and  $w_j$  together),  $C_{1,2}$  (the number of occurrences of  $w_i$  without  $w_j$ ),  $C_{2,1}$  (the number of occurrence of  $w_j$  without  $w_i$ ) and  $C_{2,2}$  (the number of bigrams in the corpus that don't contains  $w_i$  or  $w_j$ ).

**Step 2: Semantic similarity measure:** We define a score measure that allows us to choose the closest semantic tree  $ST_{S_k}$  to the tree of the orig-

inal sentence  $T_{os}$ . The score measure is defined in what follows (see equation 4).

$$\text{Score} = \sum_{N_i \in T_{os}} \left( \sum_{N_j \in ST_{S_k}} \left( \frac{wc_{ij}}{ST_{S_k}(L(N_j))} \right) / \text{Nb}(ST_{S_k}) \right) / \text{Nb}(T_{os}) \quad (4)$$

The score measure is the average of the product between nodes of  $ST_{S_k}$  and  $T_{os}$ . Where  $\text{Nb}(T_{os})$  is the total number of nodes in  $T_{os}$  and  $\text{Nb}(ST_{S_k})$  is the total number of the nodes linked to each node of  $ST_{S_k}$ .  $ST_{S_k}(L(N_j))$  corresponds to the level of the node  $N_j$  contained in the semantic tree  $ST_{S_k}$ .

As a result we give the sense that corresponds to the semantic tree that obtains the highest score.

The weights obtained by the collocation measures  $wc_{ij}$  are normalized to low weights between 0 and 1.

**Step 3: Voting procedure:** The idea is that during the classification task, we ranked measures of collocation according to the correct attribution of the sense.

In the case where the three collocation measures agree on the same result, then the given sense will be attributed to the ambiguous word and the rank of the collocation measure will not be changed.

In the case where more than one measure agrees on the attribution of a sense, then, we have to choose the sense having the majority of votes. The rank of the measures that vote for the attributed sense will be increased and the rank of the other measures will be decreased.

The final case is where all the measures do not give the same result. The result given by the measure having the highest rank (attributed during the last N tests) will be used to attribute the sense of the ambiguous word. In what follows, we detail results given by the described method.

## 3 Experimental Results

### 3.1 Used resources and tested data

Due to our need to maximize the keywords that define a specific sense, we use Arabic Wordnet (AWN) (Black et al., 2006) which is a dictionary. Words are arranged semantically instead of alphabetically. Synonymous words are grouped together to form synonym sets.

Also we collect a large corpus from newspaper articles, which were recorded from different corpus that are available on the net. In total, we collect a corpus that counts 123,854,642 words.

For the missed senses in the corpus, we collect from the net the contexts containing these senses and we added them to the used corpus.

### 3.2 Obtained Results

In the table 1 below, we report the statistics of the tested data and the obtained rate (Precision, Recall, F-Score) given by the voting procedure (VP) and the collocation measures for 127 ambiguous words. In total we test 42,316 sentences.

For each sense we test 40 sentences. For the classification part of the voting procedure, we use 20 samples per sense (labeled data). In total, we have 4,582 tagged samples. We haven't found an important difference between the sense tags, the agreement between the annotators is in the average of 95%.

$w_{c_j}$	#correct disambiguated sentences	Recall	Precision	F-Score
$T_{test}$	31,298	0,739	0,754	0,747
MI	29,783	0,703	0,718	0,710
$\chi^2$	32,122	0,759	0,774	0,766
VP	35,145	0,830	0,830	0,830

Table1. Performances of our method.

We remark that the F-score obtained by applying the voting procedure is higher than those obtained by any one of the collocation measures.

There is not a big difference between the Precision and the Recall obtained by any of the used collocation measures. This can be explained by the fact that the majority of the tested words were disambiguated. However, the best collocation measure is the  $\chi^2$ , otherwise the voting procedure increases the F-score by 6,4%.

We measure the performance of our method under the number of nodes in ST. The obtained results indicate that for semantic trees with at least 500 nodes, the performance of our method increases consistently. However, the F-Score reaches the top and becomes stable for semantic tree sizes between 2,000 and 3,000 nodes. We conjecture that more the semantic tree is enriched by the nodes, more the F-Score increases.

### 3.3 Comparison with other works

In order to contextualize the obtained results in the current state of the art, fifty ambiguous words that are used in the experimental study of this work were evaluated in previous works of Arabic WSD:

- Supervised works which are the naïve bayesian algorithm, the Decision List and the K Nearest Neighbor (Merhbene et al., 2012).
- Based knowledge works which are the original Lesk algorithm and the modified Lesk algorithm that uses Arabic Wordnet and five similarity measures (Zouaghi et al., 2011).
- Unsupervised work for Arabic WSD based on a combination between some information retrieval measures and the Lesk algorithm (Zouaghi et al., 2012) and (Merhbene et al., 2010).

Compared to our method, we note that the Lesk algorithm is limited to dictionary definitions that we use. Therefore, the absence of certain words can radically change the results. The modified Lesk algorithm using the Leacock and Chodorow measure (Leacock and Chodorow, 1998) is the most performed between based knowledge methods with a rate of Precision equal to 67,73%.

The supervised methods need an important amount of tagged data to achieve satisfactory results. They need to be applied in specific domains. The K nearest neighbor algorithm achieves the best rate of Precision (52,02%).

Finally, compared to the unsupervised method of Arabic WSD, the rate of precision is enhanced by 10% using more 117 ambiguous words.

## 4 Conclusion and future work

This paper describes a novel approach for the disambiguation of the Arabic language based on the weighted directed graph.

During the step of disambiguation, we match the tree of the sentence to be disambiguated with each semantic tree of the senses candidate. The obtained weighted directed graph uses three collocation measures that will be classified using a novel supervised voting procedure. Results show that our method achieves a very high recall and precision (83%).

In the future works, we propose to test more ambiguous words, using more tested data and resources to confirm the positive obtained results.

## References

- Agirre E. and Edmond P. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Springer, New York, NY, USA.
- Agirre E. and Sorora A. 2007. *A graph based unsupervised system for induction and classification*.

- The Fourth International Workshop on Semantic Evaluations, p.p: 346-349.
- Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A. and Fellbaum, C. 2006. *Introducing the Arabic WordNet Project*, in Proceedings of the Third International WordNet Conference, Sojka, Choi, Fellbaum and Vossen eds.
- Elloumi M. 1998. *Comparison of Strings Belonging to the Same Family*. Information Sciences, An International Journal, Elsevier Publishing Co., Amsterdam, North-Holland (Publisher), 111(1-4), p.p:49-63.
- Klapaftis I, and Manandhar S. 2008. *Word Sense Induction Using Graphs of Collocations*. In the proceeding of the 18<sup>th</sup> European Conference On Artificial Intelligence, p.p: 298-302.
- Khoja, Shereen, 1999. Stemming Arabic Text. <http://zeus.cs.pacificu.edu/shereen/research.htm>
- Leacock C. and Chodorow, M. 1998. *Combining local context and WordNet sense similarity for word sense identification*. MIT Press, Cambridge, Massachusetts, p.p: 265-283.
- Manning C. and Schütze H. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge.
- Merhbene L., Zouaghi A. and Zrigui M. 2012. *Arabic Word Sense Disambiguation*. In Proceeding of International Conference on Agents and Artificial Intelligence, Volume 1, Valencia, Spain, 22-24 January, p.p:652-655
- Merhbene L., Zouaghi A. and Zrigui M. 2012. *Lexical Disambiguation of Arabic Language: An Experimental Study*. The Journal Polibits Vol 46, pp: 49-54.
- Plaza L. and Diaz A. 2011. *Using Semantic Graphs and Word Sense Disambiguation Techniques to Improve Text Summarization*. The Procesamiento del Lenguaje Natural, p.p: 97-105.
- Zouaghi A., Merhbene L., Zrigui M. 2012. *Combination of information retrieval methods with LESK algorithm for Arabic word sense disambiguation*. Journal Article published in the Artificial Intelligence Review. Volume 38, Issue 4, DOI: 10.1007/s10462-011-9249-3; Online ISSN: 1573-7462, p.p:257-269.
- Zouaghi A., Zrigui M. and Antoniadis G. 2008. *Understanding of the Arabic spontaneous speech: A numeric modelisation*, Revue TAL VARIA.
- Zouaghi A., Merhbene L., Zrigui M. 2011. *Word Sense disambiguation for Arabic language using the variants of the Lesk algorithm*, in Proceeding of the International Conference on Artificial Intelligence (ICAI'11), Las Vegas, USA, pp: 561-567.

# Incremental Segmentation and Decoding Strategies for Simultaneous Translation

Mahsa Yarmohammadi<sup>†</sup>, Vivek K. Rangarajan Sridhar<sup>°</sup>, Srinivas Bangalore<sup>°</sup>, Baskaran Sankaran<sup>‡</sup>

<sup>†</sup>Center for Spoken Language Understanding, Oregon Health & Science University

<sup>°</sup>AT&T Labs - Research

<sup>‡</sup>School of Computing Science, Simon Fraser University

{yarmoham}@ohsu.edu, {vkumar,srini}@research.att.com, {baskaran}@cs.sfu.ca

## Abstract

Simultaneous translation is the challenging task of listening to source language speech, and at the same time, producing target language speech. Human interpreters achieve this task routinely and effortlessly, using different strategies in order to minimize the latency in producing target language. Toward modeling the human interpretation process, we propose a novel input segmentation method using the phrase alignment structure of the language pair. We compare and contrast three incremental decoding and two different input segmentation strategies, including our proposed method, for simultaneous translation. We present accuracy and latency tradeoffs for each of the decoding strategies when applied to audio lectures from the TED collection.

## 1 Introduction

In simultaneous speech translation, it is important to keep the delay between a source language chunk and its corresponding target language chunk (referred to as *ear-voice span*) minimal in order to continually engage the listeners. Simultaneous human interpreters are able to generate target speech incrementally with very low ear-voice span by using a variety of strategies (Chernov, 2004) such as anticipation, cognitive and linguistic inference, paraphrasing, etc. However, current methodologies for simultaneous translation are far from being able to exploit or model such complex phenomena. Quite often, models trained for consecutive translation are repurposed for incremental translation.

One of the first attempts at incremental *text* translation was presented by Furuse and Iida (1996) using a transfer-based MT approach and more recently by Sankaran et al. (2010) using a phrase-based approach. On the other

hand, incremental *speech* translation has been addressed in simultaneous translation of lectures and speeches (Hamon et al., 2009; Fügen et al., 2007). Some previous work (Cettolo and Federico, 2006; Rao et al., 2007; Matusov et al., 2007) addressed source text (reference or ASR hypothesis) segmentation strategies in speech translation. Constraining the search process during decoding to be monotonic (Tillmann and Ney, 2000) is one way of reducing latency and promoting incrementality. However, finding the optimal segmentation of the complete source string using dynamic programming is still slow.

By shifting the focus of the task to appropriate segmentation of incoming text, consecutive translation models have been used with good success to simulate incremental translation, such as incremental speech-to-speech translation (Bangalore et al., 2012) which focuses on translating the partial hypotheses generated based on the silences detected by a speech recognizer. However, studies on human interpreters show that in only a few cases the interpreters encode the chunks of speech as uttered in the source: the mean proportion of silence-based chunking by interpreters is 6.6% when the source is English, 10% when it is French, and 17.1% when it is German (Pöschhacker, 2002). As an alternative to silence-based segmentation, in this work, we propose a novel approach for segmenting the incoming text that exploits the alignment structure between words (phrases) across a language pair. We compare the two segmentation methods in three different decoding strategies. We perform our investigation within an English-French phrase-based speech translation system trained and tested on TED talks released as part of the IWSLT evaluation (Federico et al., 2011).

## 2 Non-incremental and Incremental Translation

The objective in machine translation is to translate a source sentence  $\mathbf{f} = f_1^J = f_1, \dots, f_J$  into target sentence  $\mathbf{e} = e_1^I = e_1, \dots, e_I$ . Given the in-

put sentence  $\mathbf{f}$ , we choose the sentence with highest probability among all possible target sentences. Since, it is intractable to estimate the conditional probability distribution  $\Pr(\mathbf{e}|\mathbf{f})$  over sentences, we simplify the problem as mapping between sentential sub-units (words or phrases) and represent the correspondence across these units using an alignment structure,  $\mathbf{a} = a_1^J = a_1, \dots, a_J$ .

$$\hat{\mathbf{e}}(\mathbf{f}) = \arg \max_{\mathbf{e}} \left\{ \sum_{\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a}|\mathbf{f}) \right\} \quad (1)$$

In an incremental translation framework, we do not observe the entire string  $\mathbf{f}$ . Instead, we observe segments of the string. A sentence pair  $(f_1^J, e_1^J)$  can be segmented into  $K$  phrase pairs  $\mathbf{s} = s_1^K = s_1, \dots, s_K$ ,

$$s_k = (i_k; b_k, j_k) \quad \forall k = 1, \dots, K \quad (2)$$

where  $i_k$  is the end position of the word in target phrase  $k$  and  $(b_k, j_k)$  represent the start and end positions of the source phrase aligned with the target phrase  $k$ . To achieve the highest *monotonicity* in incremental translation, we may restrict the decoding problem to strictly generate *monotonic phrases* by satisfying the constraint,  $b_k = j_{k-1} + 1 \quad \forall k = 1, \dots, K$ . We also constrain the source and target phrases to be ordered monotonically, meaning that if a source phrase at position  $j$  is translated to a target phrase at position  $i$ , then a source phrase at position  $j' > j$  will be translated to a target phrase at position  $i' > i$ . We call such phrase pairs to be a *monotonic phrase alignment* for a sentence pair. Figure 1 shows an example of a word alignment matrix, all possible phrase pairs, and all possible monotonic phrase alignments (4 alignments) for the parallel sentences  $\mathbf{e}$ - $\mathbf{f}$ , shown with different line styles. For instance, the monotonic phrase alignment shown with dark lines has three phrase pairs  $s_1 = (0; 0, 0)$ ,  $s_2 = (3; 1, 3)$ ,  $s_3 = (4; 4, 5)$ . Grey dotted-line phrases are not monotonic. In Section 3.2 we present a source sentence segmentation approach that makes use of the monotonic phrase alignments information.

### 3 Segmentation of ASR output for MT

In this section, we describe two alternative methods to split the input sentence into partial segments for incremental translation. Since the ASR component is not the main focus of our study, we do not explain the ASR system we used in detail. Our ASR system uses context-dependent HMMs

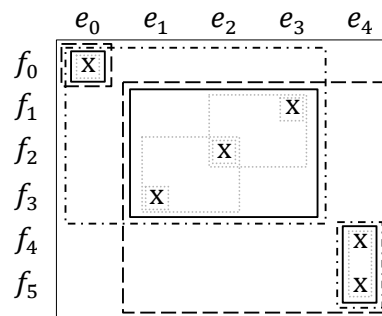


Figure 1: Word alignment matrix for two parallel sentences and their monotonic phrase alignments.

with Vocal Tract Length Normalization (VTLN) to build its acoustic model from 1119 talks we harvested from the TED website. We used the AT&T FSM toolkit (Mohri et al., 1997) to train a trigram language model for English from the permitted data in IWSLT 2011 evaluation. We reached 78.8% and 77.4% ASR word accuracies on the IWSLT dev2010 and tst2010 sets respectively.

#### 3.1 Silence-based Segmentation

The output of automatic speech recognition includes silence information that is typically discarded before passing the source string into the machine translation component. We use any silence, irrespective of the frame length, as a segmentation marker. The average length of a segment using this strategy is  $4.28 \pm 3.28$  words.

#### 3.2 Monotonic Phrase-Based Segmentation

In this section, we present an approach to split the source sentence into segments that can be monotonically translated to the target language. To prepare the training data for our segmentation model, we extracted monotonic phrase alignments from the set of all possible phrase alignments of a sentence pair in the word alignment matrix produced by GIZA++ using dynamic programming. We used 90% of the total parallel sentences and their extracted monotonic phrase alignments as training set, and reserved the rest 10% as development set. To get more meaningful alignments, we restricted those to the alignments of length at least 4.

Having the above training data, we trained a binary classifier, which was applied independently at each word in the sentence, to decide whether that word is a segment boundary or not. We used a discriminative log-linear model to train the classifier and we used the perceptron algorithm (Collins, 2002) to train the model parameters. Fisher and

Roark (2007), successfully used a discriminative log-linear model using the perceptron algorithm for automatic discourse segmentation task.

The task is to learn a mapping from inputs  $x \in X$  to outputs  $y \in Y$ , where  $X$  is the set of sentences and  $Y$  is the set of possible monotonic alignments of the sentences. Given a set of training examples  $(x_i, y_i)$ , a function  $\mathbf{GEN}(x)$  that enumerates a set of possible monotonic alignments of  $x$ ,  $\bar{\alpha} \in \mathbf{R}^d$  a parameter vector, and representation  $\Phi$  that maps each  $(x, y) \in X \times Y$  to a feature vector  $\Phi(x, y)$ , there is a mapping from an input  $x$  to an output  $F(x)$  defined by the formula:

$$F(x) = \arg \max_{y \in \mathbf{GEN}(x)} \Phi(x, y) \cdot \bar{\alpha} \quad (3)$$

The model learns the parameter values  $\bar{\alpha}$  during the training, and the decoding algorithm searches for the  $y$  that maximizes 3. The feature vector  $\Phi(x, y)$  represents arbitrary features of the alignments. In our study, the feature set contains word, position of the word in the sentence, and segment length. For example, one feature might be (word='cat', position=8, seg\_length=3, seg\_boundary = true), which returns 1 if the current word is 'cat', it is the 8th word in the sentence, it is the 3rd word in the segment, and it is marked as a segment boundary, and returns 0 otherwise.

We evaluated our segmentation model with precision, recall and F1-score, defined in Eq. 4. Suppose a sentence of length  $n$  has  $m$  segment boundaries in the gold standard and  $k$  segment boundaries in the system output. Assume  $t$  out of  $k$  guessed boundaries are correct. Since we might have multiple valid segmentations for a sentence in our training data, we chose the gold standard to be the valid segmentation which has the minimum Levenshtein edit distance with the system output.

$$P = \frac{t}{k}, R = \frac{t}{m}, F1 = \frac{2PR}{P+R} = \frac{2t}{k+m} \quad (4)$$

We achieved  $P = 70.51\%$ ,  $R = 91.52\%$ , and  $F1 = 75.89\%$  on the development set. The average length of a segment using this strategy is  $6.56 \pm 4.73$  words.

## 4 Decoding Strategies

We used three different decoding strategies for translating the ASR outputs. We tried each of these three techniques for incremental as well as regular (non-incremental) translation.

First, we used the Moses toolkit (Koehn et al., 2007) for statistical machine translation. Minimum error rate training (MERT) was performed on the development set (dev2010) to optimize the feature weights of the log-linear model used in translation. During decoding, the unknown words were preserved in the hypotheses. The parallel text for building the English-French translation model – around 6.3 million parallel sentences – was obtained from several corpora: Europarl (Koehn, 2005), jrc-acquis corpus (Steinberger et al., 2006), Opensubtitle corpus (Tiedemann and Lars Nygaard, 2004), WMT11 Gigaword (Callison-Burch et al., 2011), WMT11 News (Callison-Burch et al., 2011), and Web crawling (Rangarajan Sridhar et al., 2011) as well as human translation of proprietary data.

Second, we used a finite-state implementation of translation without reordering. We represent the phrase translation table as a weighted finite state transducer (FST) and the language model as a finite-state acceptor. The weight on the arcs of the FST is the dot product of the MERT weights with the translation scores. Our FST-based translation is the equivalent of phrase-based translation in Moses without reordering.

In addition to Moses and FST decoders, we used the incremental beam search decoder introduced by Sankaran et al. (2010) for translating in regular and incremental modes. This decoder modifies the beam-search decoding algorithm for phrase-based MT aiming at efficient computation of future costs and avoiding search errors. In Section 6 we show the results of translating our data using these three decoding strategies, referred to as Moses, FST and IncBeam decoders.

## 5 Data

In this work, we focus on the speech translation of TED talks. Over the past couple of years, the International Workshop on Spoken Language Translation (IWSLT) has been conducting the evaluation of speech translation on TED talks for English-French. We leverage the IWSLT TED campaign by using identical development (dev2010) and test data (tst2010).

## 6 Experiments and Results

We compare the results in terms of accuracy of translation and latency of generating partial outputs. We translated and evaluated each of 11 test

sets independently and we report the average values. In incremental mode, we ran Moses with *continue-partial-translation* option which enables chunk translation to be conditioned on history. In contrast, FST performs a chunk-wise translation which is independent of history.

		Moses	FST	IncBeam
Regular	ASR	18.67	18.11	17.73
	Transcript	22.66	22.11	21.32
Incr. silence seg.	ASR	17.41	16.88	17.33
Incr. monotone seg.	ASR	17.64	17.09	17.40

a) Reference translation has punctuations

		Moses	FST	IncBeam
Regular	ASR	23.04	22.58	22.00
	Transcript	28.38	27.75	26.63
Incr. silence seg.	ASR	21.66	21.12	21.38
Incr. monotone seg.	ASR	21.69	21.26	21.48

b) Reference translation has no punctuations

Table 1: Accuracy (BLEU) of English-French MT models on reference transcripts and ASR outputs

Table 1 shows translation accuracies in terms of BLEU scores. We consider the regular decoding as the baseline. Since we know the entire source input in advance, our baseline, obviously, has the highest accuracy but also the highest latency. For the baseline, we translated the ASR output and the reference transcript of the utterance. As shown in the "Regular" row, the accuracy on the ASR output drops by around 4% compared to that on the reference text. Since ASR outputs and the training data for our translation model do not contain punctuations, we also measured the accuracy against the references with removed punctuations.

Incremental translation of monotone-based segments gets a slightly higher accuracy than the silence-based segments for all the three decoders. In both regular and incremental decoding settings, the BLEU scores of Moses are higher than other two decoders. The FST decoder is better than the IncBeam decoder in regular setting; on the other hand the performance of the IncBeam decoder is better than the FST decoder and comparable to Moses in the two incremental settings. Both Moses and IncBeam decoders use reordering knowledge as well as history of translation in the incremental decoding settings, whereas the FST decoder lacks the latter.

In Table 2, we present the average speed of translating ASR output chunks. For each sentence the speed is calculated as the total time taken to translate the chunks divided by the number of

	Moses	FST	IncBeam
Regular	2.35	2.06	17.68
Incr. silence seg.	0.68	1.75	6.43
Incr. monotone seg.	0.87	1.59	8.60

Table 2: Speed of generating target chunks (sec)

chunks of the sentence. The speed reported in the table is then calculated by taking the average of speeds of all sentences in the test set. This measurement provides a good indication of latency in real-time translation. We note that we do not compare the delay of the decoders with each other due to differences in implementation and invoking the decoders, instead we compare the delays of each decoder by itself in three modes of translation.

Comparing the accuracy values in Table 1 and latency values in Table 2 shows that in incremental decoding using the Moses and IncBeam decoders, we get some gain in accuracy but we lose some speed in monotone-based model compared to the silence-based model.

The interesting achievement is in incremental translation of monotone-based segments using the FST decoder. In this condition, we not only achieve an improvement in accuracy, but we also get a reduction in latency compared to the translation of silence-based segments. When translating each chunk independently, a meaningful segmentation of the input toward increasing the monotonicity yields a better performance in simultaneous translation than a silence-based segmentation.

## 7 Conclusions

In this paper we introduced a novel incoming text segmentation approach aiming at increasing the monotonicity of simultaneous translation. Using our proposed framework, we could achieve a point in segmenting and decoding the ASR output which enables simultaneous speech translation with a good accuracy/latency trade-off, even without relying on the history of translation. For future work we plan to improve our monotone-based segmentation model by using richer feature sets which for example include syntactic knowledge of the language. We are also interested in exploring our techniques on translating the languages with different word orders such as English/Japanese.



## Acknowledgments

We would like to thank Brian Roark for his valuable discussions.

## References

- S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of NAACL:HLT*, June.
- C. Callison-Burch, P. Koehn, C. Monz, and O. Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.
- M. Cettolo and M. Federico. 2006. Text segmentation criteria for statistical machine translation. In *Proceedings of the 5th international conference on Advances in Natural Language Processing*.
- G. V. Chernov. 2004. *Inference and anticipation in simultaneous interpreting*. John Benjamins.
- M. Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.
- M. Federico, L. Bentivogli, M. Paul, and S. Stüker. 2011. Overview of the IWSLT 2011 evaluation campaign. In *Proceedings of IWSLT*.
- S. Fisher and B. Roark. 2007. The utility of parse-derived features for automatic discourse segmentation. In *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 488–495.
- C. Fügen, A. Waibel, and M. Kolss. 2007. Simultaneous translation of lectures and speeches. *Machine Translation*, 21:209–252.
- O. Furuse and H. Iida. 1996. Incremental translation utilizing constituent boundary patterns. In *In Proc. of Coling '96*, pages 412–417.
- O. Hamon, C. Fügen, D. Mostefa, V. Arranz, M. Kolss, A. Waibel, and K. Choukri. 2009. End-to-end evaluation in simultaneous translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, March.
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. J. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*.
- P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.
- E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tür, M. Ostendorf, and H. Ney. 2007. Improving speech translation with automatic boundary prediction. In *Proceedings of Interspeech*.
- M. Mohri, F. Pereira, and M. Riley. 1997. Att general-purpose finite-state machine software tools, <http://www.research.att.com/sw/tools/fsm/>.
- F. Pöchhacker. 2002. *The Interpreting Studies Reader*. Routledge (Taylor and Francis), New York.
- V. K. Rangarajan Sridhar, L. Barbosa, and Bangalore. S. 2011. A scalable approach to building a parallel corpus from the web. In *INTERSPEECH*, pages 2113–2116.
- S. Rao, I. Lane, and T. Schultz. 2007. Optimizing sentence segmentation for spoken language translation. In *Proceedings of Interspeech*.
- B. Sankaran, A. Grewal, and A. Sarkar. 2010. Incremental decoding for phrase-based statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.
- R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, and D. Tufis. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of LREC*.
- J. Tiedemann and L. Lars Nygaard. 2004. The OPUS corpus - parallel & free. In *Proceedings of LREC*.
- C. Tillmann and H. Ney. 2000. Word re-ordering and dp-based search in statistical machine translation. In *In Proc. of the COLING 2000, JulyAugust*, pages 850–856.

# Two Case Studies on Translating Pronouns in a Deep Syntax Framework

Michal Novák, Zdeněk Žabokrtský and Anna Nedoluzhko  
Charles University in Prague, Faculty of Mathematics and Physics  
Institute of Formal and Applied Linguistics  
Malostranské náměstí 25, CZ-11800  
{mnovak, zabokrtsky, nedoluzko}@ufal.mff.cuni.cz

## Abstract

We focus on improving the translation of the English pronoun *it* and English reflexive pronouns in an English-Czech syntax-based machine translation framework. Our evaluation both from intrinsic and extrinsic perspective shows that adding specialized syntactic and coreference-related features leads to an improvement in translation quality.

## 1 Introduction

Machine Translation (MT) is an extremely broad task and can be decomposed along various directions. One of them lies in using specialized translation models (TMs) for certain types of language expressions. For instance, different types of named entities often receive specialized treatment in real translation systems. This paper deals with introducing specialized TMs for two types of pronouns: the pronoun *it* and reflexive pronouns. The models are integrated into an English-Czech syntax-based MT framework.

Several works have previously focused on translating pronouns. The linguistic study of Morin (2009) investigated the translation of pronouns, proper names and kinship terms from Indonesian into English. Onderková (2010) has conducted a corpus-based research on possessive pronouns in Czech and English, focusing especially on their use with parts of the human body.

From the perspective of MT, translating personal pronouns from English to morphologically richer languages, such as French (Le Nagard and Koehn, 2010), German (Hardmeier and Federico, 2010) and Czech (Guillou, 2012) has recently aroused higher interest. In these languages, one usually has to ensure agreement in gender and number between the pronoun and its direct antecedent, which requires a coreference resolver to be involved.

In this work, we make use of the English-to-Czech translation implemented within the TectoMT system (Žabokrtský et al., 2008). In contrast to the phrase-based approach (Koehn et al., 2003), TectoMT performs a tree-to-tree machine translation. An input English sentence is first analyzed into its deep-syntactic representation, which is subsequently transferred into Czech. The pipeline ends with generating a surface form of the Czech translation from its deep representation.

The deep syntactic representation of a sentence in TectoMT follows the Prague tectogramatics theory (Sgall, 1967; Sgall et al., 1986). It is a dependency tree whose nodes correspond to content words. Personal pronouns missing on the surface are reconstructed in special nodes. All nodes are assigned semantic roles and coreference relations are annotated.

Originally, translation of both *it* and reflexive pronouns was treated by rules in TectoMT. The English deep representation of *it* was translated as *to* and a simple heuristics determined if it is being expressed on the surface. Similarly, reflexives were always translated as *se*. This paper evaluates the translation quality reached using specialized classifiers for these pronouns. Unlike the related work on pronouns in MT, we focus on improving the lexical choice, not tuning other components that affect generating a particular surface form (e.g. coreference resolution).

## 2 Linguistic analysis

We started with an analysis of how the pronouns under investigation are translated<sup>1</sup> in two Czech-English parallel treebanks – Prague Czech-English Dependency Treebank 2.0 (Hajič et al., 2011, PCEDT) and CzEng 1.0 (Bojar et al., 2012).

<sup>1</sup>Note that besides the means mentioned below, there are other ways of translating these pronouns. However, in most cases they can be replaced by one of the variants listed with no harm to the quality of the Czech output.

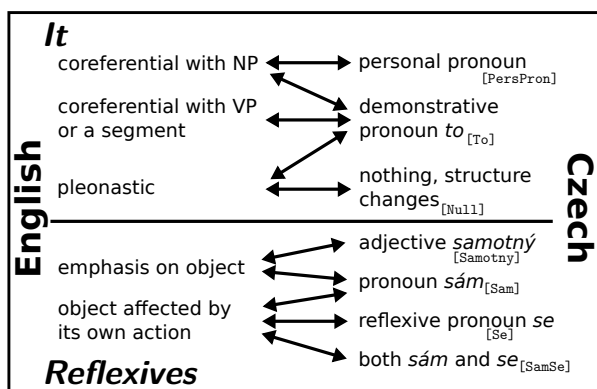


Figure 1: The mapping of the types of English *it* (top) and reflexive pronouns (bottom) to their Czech counterparts.

## 2.1 Translating *it* from English to Czech

In English, three coarse-grained types of *it* are traditionally distinguished: referential *it* pointing to a noun phrase in the preceding or following context, anaphoric *it* referring to a verbal phrase or a larger discourse segment, and non-referential pleonastic *it*, whose presence is imposed only by the syntactic rules of English.

There are three prevailing ways of translating *it* into Czech, also three different ways prevail. Personal pronouns or zero forms<sup>2</sup>, whose gender and number are determined by their antecedent, are the most frequent variant (referred to as the `PersPron` class in the following). Another way is using the Czech demonstrative pronoun *to*, which is a neuter singular form of the pronoun *ten* (`To` class). The third option results in fact no lexical counterpart in the Czech translation, the English and Czech sentences thus having a different syntactic structure (`Null` class).

The mapping between English and Czech types is shown in Figure 1. The `To` class is particularly overloaded. Even if a given occurrence of *it* corefers with a noun phrase, translating it to *to* does not require identifying the antecedent since the gender and number of *to* are always fixed (see Example 1).

- (1) Some investors say Friday’s sell-off was a good thing. “*It* was a healthy cleansing,” says Michael Holland.

Někteří investoři říkají, že páteční výprodej byla dobrá věc. “Byla *to* zdravá očista,” říká Michael Holland.

<sup>2</sup>Czech is a pro-drop language.

## 2.2 Translating reflexive pronouns from English to Czech

According to the *Longman Dictionary of Contemporary English*,<sup>3</sup> reflexive pronouns are typically used in two scenarios: to show that the object is affected by its own action and to emphasize that the utterance relates to one particular thing, person etc. (see Example 2).

- (2) The Gambia’s President *himself* participated in the hunt last year.

The most usual Czech counterparts of English reflexives comprise the Czech reflexive pronoun *se* (`Se` class), the adjective *samotný* (`Samotny` class) and the pronoun *sám* (`Sam` class), all in various morphological forms. Moreover, *sám* often appears with *se* to emphasize that the action affecting the object is performed by the object itself (`SamSe` class). Figure 1 illustrates the correspondence between English usages and Czech expressions.

## 3 Data

To train and intrinsically evaluate TMs for *it* and English reflexives, we have extracted data from the entire PCEDT and 11 sections of CzEng. Both treebanks follow the annotation style based on the Prague tectogramatics theory (Sgall, 1967; Sgall et al., 1986). While PCEDT consists of 50,000 sentence pairs annotated mostly manually, the annotation of CzEng with 15 million parallel sentences is entirely automatic. Both treebanks have been provided with a fully automatic alignment of Czech and English nodes (Mareček et al., 2008), which is, however, prone to errors for *it* and its Czech counterparts. Since they are pronouns, they can replace a wide range of content words and their meaning is inferred mainly from the context. The situation is better for verbs as their usual parents in dependency trees: since they carry meaning in a greater extent, their automatic alignment is of a higher quality.

We took advantage of this property and the gold annotation of semantic roles in PCEDT, obtaining Czech translations as the argument of the Czech verb aligned with the English parent verb that fills the same semantic role as the given *it*. Using this approach, we succeeded in reaching the Czech counterpart in more than 60% of instances. The rest had to be done manually.

<sup>3</sup><http://www.ldoceonline.com>

<i>It</i>	Train	Test	Reflexives	Train	Test
PCEDT sections	00–19	20–21	CzEng sections	00–09	98
PersPron	576	322	Se	6,305	652
To	231	138	Sam	2,271	205
Null	133	83	SamSe	1,361	129
			Samotny	804	89
Total	940	543	Total	10,741	1,075

Table 1: Distribution of classes in the data sets.

Czech counterparts of English reflexive pronouns have been collected directly from the alignment in CzEng, ignoring the cases where the aligned Czech word does not fall in one of the classes mentioned in Section 2.2.

The overall statistics of the train and the test set are shown in Table 1. The disproportion of training instances for *it* results from the manual annotation of classes, which could not be completely finished due to time reasons. In order to maintain the overall distribution, we also had to limit the number of automatically annotated classes.

Given the observation (see Section 2), we designed features to differentiate between the ways *it* and reflexives are translated.

### 3.1 Features for *it*

The translation mapping in Figure 1 suggests that identifying the English type of *it* might be informative. We thus constructed a binary coreference-related feature based on the output of the system NADA (Bergsma and Yarowsky, 2011) giving an estimate of whether an instance of *it* is coreferential.

Some verbs are more likely to bind with *it* that refers to a longer utterance. Such *it* is relatively consistently translated as a demonstrative *to*. However, PCEDT is too small to be a sufficient sample from a distribution over lexical properties. Hence, we took advantage of CzEng and collected co-occurrence counts between a semantic role that the given *it* fills concatenated with a lemma of its verbal parent and a Czech counterpart having the same semantic role (denoted as *csit*). We filtered out all occurrences where *csit* was neither a personal pronoun nor *to*. For both possible values of *csit* a feature is constructed by looking up frequencies for a concrete occurrence in the co-occurrence counts collected on CzEng and quantized into 4–5 bins following the formula:

$$\text{bin}\left(\log\left(\frac{\text{count}(\text{semrole} : \text{parent} \wedge \text{csit})}{\text{count}(\text{semrole} : \text{parent})\text{count}(\text{csit})}\right)\right).$$

Linguistic analysis suggested including syntax-

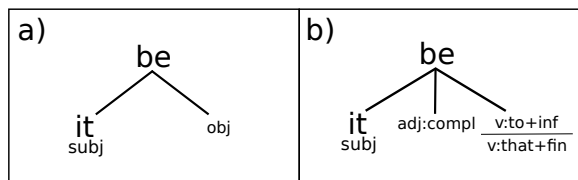


Figure 2: Examples of syntactic features capturing typical constructions with a verb *be*.

oriented patterns related to the verb *to be* such as those shown in Figure 2. For instance, nominal predicates<sup>4</sup> tend to be translated as *to* even if *it* is coreferential. On the other hand, an adjectival predicate followed by a subordinating clause introduced by the English connectives *to* or *that* usually indicates a pleonastic usage of *it* translated as a null subject.

### 3.2 Features for reflexive pronouns

Here we focused on distinguishing between the two most frequent meanings (see Section 2.2). Ideally, the POS tag of the parent would be a sufficient feature because reflexives in the second meaning should depend on a noun. However, since we deal with automatically parsed trees we had to support the parent POS tag by the POS tag of the immediately preceding word. Moreover, another feature indicates if the preceding word is a noun and agrees with the pronoun in gender and number.

Furthermore, we observed that *sám* rarely appears in other case than nominative. Although this feature exploits the target side, we can use it since the case of the governing Czech noun is already known at the point when reflexives are translated.

Last but not least, the morpho-syntactic pattern (including a possible preposition) in which the reflexive pronoun appears is a valuable feature.

## 4 Experiments and Evaluation

To mitigate a possible error caused by a wrong classifier choice, we built several models based on various Machine Learning classification methods including Maximum Entropy implemented in the AI::MaxEntropy Perl library,<sup>5</sup> logistic regression with one-against-all strategy from Vowpal Wabbit<sup>6</sup> as well as decision trees, k-NN and SVM from Scikit-learn library (Pedregosa et al., 2011).

<sup>4</sup>The verb *to be* has an object.

<sup>5</sup><http://search.cpan.org/~laye/AI-MaxEntropy-0.20>

<sup>6</sup><http://hunch.net/~vw>

	<i>It</i>		Reflexives	
	Train	Test	Train	Test
Baseline	60.70	59.30	58.70	60.65
AI::MaxEntropy	85.99	<b>76.61</b>	76.37	77.77
VW (passes=20, l2=10e-5)	89.99	76.43	76.98	77.77
sklearn:decision-trees	93.36	73.66	81.78	76.37
sklearn:k-NN (k=10)	82.51	73.30	77.64	76.74
sklearn:SVM (kernel=linear)	90.83	75.51	76.55	<b>78.14</b>

Table 2: Accuracy of both translation models on the training and test data.

We compare our results with a majority class baseline (`PersPron` and `Se` classes) in Table 2. The results show a 17% gain when our approach is used.

The specialized models have been integrated in the TectoMT system and extrinsically evaluated on the English-Czech test set for the WMT 2011 Translation Task (Callison-Burch et al., 2011).<sup>7</sup> This data set contains 3,003 English sentences with one Czech reference translation, out of which 430 contain at least one occurrence of *it* and 52 contain a reflexive pronoun.

The new approach was compared to the original TectoMT rule-based pronoun handling heuristics (see Section 1). The shift from the original settings to the new translation models results in 166 changed sentences with *it* and 17 changed sentences with English reflexives. In terms of BLEU score, we observe a marginal drop from 0.1404 to 0.1403 using the new approach. However, BLEU may be too coarse for this kind of experiment.

In order to give a more realistic view, we carried out a manual evaluation. All 17 modified sentences for reflexives and 50 randomly sampled changed sentences containing *it* were presented to one annotator who assessed which of the two systems gave a better translation. Table 3 shows that improved sentences dominate in both cases. Overall, the improved sentences account for around 8.5% of all sentences with *it* and 23% sentences containing a reflexive pronoun.

## 5 Discussion

Looking into the types of improvements and errors in the manually evaluated sentences, we have found that the new model for *it* opted for a different translation only in cases where the original system decided to express *to* on the surface. In 13 out of 24 improvements, the new model for *it* succeeded in correctly resolving the `Null` class

<sup>7</sup><http://www.statmt.org/wmt11/test.tgz>

	<i>It</i>	Reflexives
new better than old	24	12
old better than new	13	0
equal quality	13	5

Table 3: The results of manual evaluation on sentences translated by TectoMT in the original settings and using the new translation models

while in the remaining 11 cases, the corrected class was `PersPron`. It took advantage mostly of the syntax-based features in the former case and the coreference-related feature in the latter.

Regarding the reflexive pronouns the pronoun was used in its emphasizing meaning in all but two altered sentences. This accords with the design of features, which are mainly targeted at revealing this usage of reflexives. Moreover, the feature indicating if a Czech noun is in nominative case has proved to be particularly useful, correctly driving the lexical choice between *sám* and *samotný*. The majority of errors stem from incorrect activation of syntactic features due to parsing and POS tagging errors.

## 6 Conclusions

In this work, we presented specialized translation models for two types of English pronouns: *it* and reflexives. Integrating them into an English-Czech syntax-based MT system TectoMT we succeeded in improving the concerned sentences measured by human evaluation.

Generally, it is intractable to design a specific feature set for every word. However, this work shows on two examples that the correct translation of some words depends on many linguistic aspects, e.g. syntax and coreference and that is worth taking these aspects into account.

## Acknowledgments

This work has been supported by the Grant Agency of the Czech Republic (grants P406/12/0658 and P406/2010/0875), the grant GAUK 4226/2011 and EU FP7 project Khresmoi (contract no. 257528). This work has been using language resources developed and/or stored and/or distributed by the LINDAT-Clarin project of the Ministry of Education of the Czech Republic (project LM2010013).

## References

- Shane Bergsma and David Yarowsky. 2011. NADA: A Robust System for Non-Referential Pronoun Detection. In *DAARC*, pages 12–23, Faro, Portugal, October.
- Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. 2012. The Joy of Parallelism with CzEng 1.0. In *Proceedings of LREC 2012*, Istanbul, Turkey, May. ELRA, European Language Resources Association.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.
- Liane Guillou. 2012. Improving Pronoun Translation for Statistical Machine Translation. In *Proceedings of the Student Research Workshop at the 13th Conference of the EACL*, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Čínková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2011. Prague Czech-English Dependency Treebank 2.0.
- Christian Hardmeier and Marcello Federico. 2010. Modelling Pronominal Anaphora in Statistical Machine Translation. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 283–289.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the NAACL HLT – Volume 1*, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ronan Le Nagard and Philipp Koehn. 2010. Aiding 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.
- David Mareček, Zdeněk Žabokrtský, and Václav Novák. 2008. Automatic Alignment of Czech and English Deep Syntactic Dependency Trees. In *Proceedings of the Twelfth EAMT Conference*, pages 102–111.
- Izak Morin. 2009. Translating Pronouns, Proper Names and Kinship Terms from Indonesian into English and vice versa. *TEFLIN Journal: A publication on the teaching and learning of English*, 16(2).
- Kristýna Onderková. 2010. Possessive Pronouns in English and Czech Works of Fiction, Their Use with Parts of Human Body and Translation. Master’s thesis, Masaryk University, Faculty of Arts.
- 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, November.
- Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht.
- Petr Sgall. 1967. *Generativní popis jazyka a česká deklinace*. Academia, Prague, Czech Republic.
- Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular MT System with Tectogramatics Used as Transfer Layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Bootstrapping Phrase-based Statistical Machine Translation via WSD Integration

Hien Vu Huy<sup>†,‡</sup>, Phuong-Thai Nguyen<sup>†,‡</sup>, Tung-Lam Nguyen<sup>†,‡</sup> and M.L Nguyen<sup>†,‡</sup>

<sup>†</sup> University of Engineering and Technology, VNU Hanoi  
{hienvuhuy, thainp, lamnt\_52}@vnu.edu.vn

<sup>‡</sup> Japan Advanced Institute of Science and Technology (JAIST)  
nguyenml@jaist.ac.jp

## Abstract

Beside the word order problem, word choice is another major obstacle for machine translation. Though phrase-based statistical machine translation (SMT) has an advantage of word choice based on local context, exploiting larger context is an interesting research topic. Recently, there have been a number of studies on integrating word sense disambiguation (WSD) into phrase-based SMT. The WSD score has been used as a feature of translation. In this paper, we will show that by bootstrapping WSD models using unlabeled data, we can bootstrap an SMT system. Our experiments on English-Vietnamese translation showed that BLEU scores have been improved significantly.

## 1 Introduction

Conventional phrase-based systems use local context information from phrase table and language model. Though phrase based SMT achieves a jump in translation quality in comparison with word based SMT, there are still cases in which local context cannot capture correctly the meanings of source words. WSD can use features from much larger contexts and those features can overlap each other. The idea of integrating WSD into SMT rises naturally from this perspective. Previously, Varea et al. (2001) directly used context sensitive lexical models, applying these models for re-ranking n-best for their word-based maximum entropy model (MEM) SMT and achieving slight improvements in translation quality.

Chan et al. (2007) made use of WSD for hierarchical phrase-based translation for Chinese-English by utilizing two new WSD features for SMT and proposing an algorithm for scoring synchronous rules. Phrases which do not exceed a

length of two were computed WSD models. Their experiments showed that WSD can improve SMT significantly.

Simultaneously with Chan et al. (2007), Carpuat and Wu (2007) used a similar approach to the problem. The main difference was that they focused on conventional phrase-based SMT in Koehn et al. (2003) and used only one WSD feature for SMT. The limit of phrase length was the same as the value used by their SMT system. Their experiments led to the same conclusion: WSD can improve SMT.

However, approaches based on statistic frequently against deficiencies of parallel and specific domain corpora. Only a few popular languages are derived continuous financial support and interest of researchers. Therefore, it becomes an immense obstacle to apply these approaches for the remaining languages.

Recently, there are several approaches to address this impediment. Ambati et al. (2011) applied multi-strategy methods in active learning for machine translation by combining several techniques in sentence selection process. They attained significantly results while parallel training data was scarce.

In this paper, we present our study on this topic. First, by integrating WSD as a model of SMT system as shown in the Figure 1, we present how we use WSD for SMT. Then we demonstrate a method to bootstrap WSD models by using unlabelled data. Finally, we show our experimental results. We analyse various settings of WSD-SMT integration. Our results give a thorough view into the problem.

## 2 WSD for SMT

### 2.1 WSD Task

In order to use WSD for SMT, the precondition is that training data must be large enough.

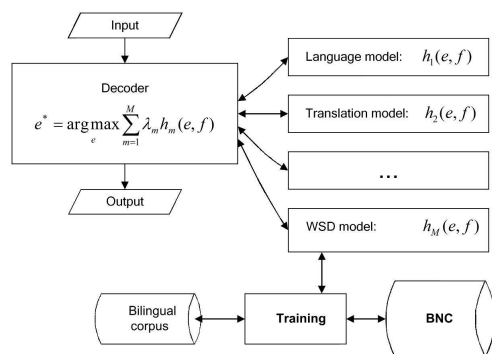


Figure 1: Integrating WSD into phrase-based SMT system

Manually-created data sets such as SENSEVAL and SemCor, which are often used in WSD studies, are too small for applications like machine translation. We overcome this difficulty by using an approach based on Carpuat and Wu (2007) and Chan et al. (2007) to extract training data from bilingual data. Word alignment information serves as a map between source words and target words. Target words are seen as senses. Since word alignment usually performs incorrectly, the resulting WSD training data is noisy. When carrying out this research, we consider WSD for word and phrase levels.

## 2.2 WSD Training Data Generation

A procedure for WSD-training-data extraction:

Input: a bilingual corpus, a POS-tagged version of the source text and word alignment information.

Output: WSD training sets for source phrases.

- Step 1: Collect phrase pair instances associated with position in the bilingual corpus. Group phrase pairs according to source phrase.
- Step 2: For each group, generate a training set for its corresponding source phrase.

Phrase pairs (s,t) which are consistent with the word alignment will be generated. The criteria of consistence with word alignment in Koehn et al. (2003) are as follows: First, there exists links from words of s to words of t. Second, for every word outside s, there is no link to any word of t. Third, for every word outside t, there is no link to any word of s.

When extracting WSD training data from a bilingual corpus, the number of training sets resulting from the extractive procedure is often much larger than vocabulary size of the source

text. Additionally, raw data extracted from a bilingual corpus is a miscellany of semantic, lexical, morphological, an syntactic ingredients. It is very different from conventional WSD data style. This data can be refined in several ways such as lemmatization.

## 2.3 WSD Features

In our work, we use six kinds of knowledge and represent them as subsets of features, as follows:

- *bag-of-words*,  $F_1(l, r) = \{w_{-l}, \dots, w_{+r}\}$ : We investigate three sets of this knowledge including  $F_1^a = F_1(-5, +5)$ ,  $F_1^b = F_1(-10, +10)$ ,  $F_1^c = F_1(-100, +100)$ , corresponding to small, medium and large size respectively.
- *collocation of words*,  $F_2 = \{w_{-l} \dots w_{+r}\}$ : As a result of the work in Le and Shimazu (2004) we choose such collocations that their lengths (including the target words) are less than or equal to 4, it means  $(l + r + 1) \leq 4$ .
- *ordered words*,  $F_3 = \{w_i | i = -l, \dots, +r\}$ : We choose  $l = r = 3$
- *collocation of POSs*,  $F_4 = \{p_{-l} \dots p_{+r}\}$ : Like collocation of words, we choose their lengths including the target words are less than or equal to 4.
- *ordered POSs*:  $F_5 = \{p_i | i = -l, \dots, +r\}$ : We choose  $l = r = 3$

In cases that we are working with a training set of a source phrase, features will be extracted from surrounding context of that phrase.

## 2.4 Integration

After having been trained, WSD models can be used as a feature for SMT as shown in the Figure 1. Since we use a log linear translation model, the use of a new feature is easy. Feature's weight is tuned using minimum error rate training (MERT) in Och (2003). In decoding phase, when translation options are generated, their WSD score is computed and then can be used in searching process. Among other features, this new feature is sensitive to large contexts.

Given a source phrase, the simplest way is to train its own WSD model and then apply that model in new contexts. The number of WSD models is equal to the number of source phrases in the SMT phrase table. An alternative is to score a phrase using shorter phrases. That means only WSD models for phrases whose length is smaller than a threshold to be trained. This setting could



reduce computational time. Suppose that we are considering a phrase pair  $(s, t)$  in which  $s$  is a source phrase,  $t$  is a target phrase. If this phrase pair can be split into a sequence  $(s_i, t_i)$  of  $n$  sub phrase pairs which are consistent with the word alignment of  $(s, t)$ , then the probability of  $t$  given  $s$  and its context can be computed using (1) here

$$P_{wsd}(t|s) \approx \prod_{i=1}^n P_{wsd}(t_i|s_i) \quad (1)$$

$P_{wsd}(t_i|s_i)$  calculates the probability of  $t_i$  conditioning on  $s_i$  and its surrounding context. If there are more than one possible split, we use a greedy method. This method gives preferences to sub phrases according to their length and score.

### 3 Using Unlabelled Data

#### 3.1 Basic Algorithm

Suppose that we have two data sets, one labelled (eg., the data extracted from a bilingual corpus) and the other unlabelled. First, a classifier is trained using the labelled data set, then it can be used to classify the unlabelled data set. Among newly labelled examples, the ones with high score will be chosen to enlarge the training data. These steps are repeated until a stopping condition is matched. Stopping condition can be a maximum number of iterations, or a minimum increase in classification accuracy, etc.

Input:  $L$  = a labelled data set.

$U$  = an unlabelled data set.

Output:  $L_{new}$ , a new labelled data set.

1. Train a classifier  $C$  using  $L$ .
2. For each  $u \in U$ :
  - a. use  $C$  to classify  $u$ .
  - b. find the label assigned with highest score.
  - c. if the score is above a threshold, choose  $u$ .
3.  $L_{new} = L \cup \{u \in U : u \text{ has been labelled}\}$ .  
and  $U_{new} = \{u \in U : u \text{ unlabelled}\}$ .
4. If the stopping condition is not matched, repeat from step 1, else stop.

#### 3.2 A New Algorithm with Sense Distribution Control

A problem with the basic algorithm is that after extension, the resulting labelled data set can be highly imbalanced in sense distribution with dominating senses, due to which the classification accuracy decreases. To handle the problem, the change of sense distribution during extending process should be controlled. We propose to use the

relative entropy or Kullback-Leibler distance in Cover and Thomas (2006) to measure the change in sense distribution and control the amount of new examples. After extending using the previous algorithm, we will remove examples one by one until the KL distance is smaller than a threshold. The threshold need not to be a fixed number.

Algorithm: Input: a labelled data set  $L_{initial}$  and its expanded set  $L_{new}$ .

Output: a labelled data set  $L_{extending}$  whose sense distribution is controlled

1.  $p = (p_1, p_2, \dots, p_n)$  and  $q = (q_1, q_2, \dots, q_n)$  are sense distributions over  $L_{initial}$  and  $L_{new}$ .
2. Compute the Kullback Leibler distance between  $p$  and  $q$ :  $\Delta = KL(p, q) = \sum_i^n p_i \log(\frac{p_i}{q_i})$
3. Repeat
  - a. for each  $u \in L_{new}$ :
    - compute  $t = (t_1, t_2, \dots, t_n)$ , the sense distribution over  $T = L_{new} \setminus \{u\}$ , then compute  $KL(t, p)$
    - find  $u_m$  minimizing  $KL(t, p)$ , then  $u_m$  is the element that when removing it, the KL distance decreases a maximum amount.
  - b. Remove  $u_m$  from  $L_{new}$
  - c. Compute  $KL(p, q)$
4. The iteration stop when  $KL(p, q) < \frac{\Delta}{2}$
5.  $L_{extending} = L_{new}$ .
6. Return  $L_{extending}$ .

### 4 Evaluation

#### 4.1 Corpora and Tools

The corpus in our experiments is English-Vietnamese bilingual corpus from several different fields which includes approximately 135,000 sentence pairs. It is divided into three parts: training, developing and testing in Table 1. We used the developing set in the evaluation of MERT of SMT system in all experiments. In addition to the testing set extracted from the bilingual corpus, we used an additional corpus consisting of ambiguous words that are labelled by evaluators to test the external domain. The rate of Out-of-Vocabulary in testing sets is roughly 2%.

In our experiments, the British National Corpus (BNC) in Clear (1993) has been used for our expansion. We used a word-segmentation program in Nguyen et al. (2003), Moses in Koehn et al. (2007), GIZA++ in Och and Ney (2000), SRILM in Stolcke (2002), a rule-based morpho-

logical analyser in Pham et al. (2003) and Natural Language Toolkit in (Bird et al., 2009) for segmenting Vietnamese sentences, learning phrase translations, creating word alignment, learning language models, analysing morphology and exploiting BNC respectively.

	Number of sentences	Average length of sentences	Number of words
<b>Training corpus</b>			
English	131,118	15.9	2,096,073
Vietnamese	131,118	17.0	2,236,847
<b>Developing corpus</b>			
English	218	15.4	3,367
Vietnamese	218	16.5	3,609
<b>Testing corpus</b>			
English	2,000	17.8	35,797
Vietnamese	2,000	19.4	38,814
<b>External-domain testing corpus</b>			
English	123	18.7	2,308

Table 1: Statistics for training, testing and developing corpora

## 4.2 Experiments and results

	Without WSD	WSD integration	WSD integration with BNC
BLEU	34.93	35.43	36.47
NIST	7.4491	7.4937	7.7971

Table 2: BLEU scores of SMT based on phrase-based with WSD and BNC-extended WSD

As indicated from the Table 2, that SMT system utilizes WSD with expanded information of BNC corpus leads to the high translation quality with growths by 1.04 and 1.54 in BLUE score and 0.3034 and 0.3488 in NIST score in comparison with non-extended WSD integrated SMT system and baseline SMT system. Let consider the example:

Input: *hard water is water that has high mineral content (in contrast with soft water).*

SMT: **chăm\_chi**/(hard) nước/(water) là/(is) nước/(water) cao/(high) nội\_dung/(content) khoáng\_sản/(mineral) trái/(in contrast) với/(with) nước/(water) mềm/(soft) .

SMT + WSD: **khó**/(hard) nước/(water) là/(is) nước/(water) có/(has) hàm\_lượng/(content) khoáng\_sản/(mineral) cao/(high) mềm/(soft) (ngược\_lại)/(in contrast) với/(with) nước/(water).

SMT + WSD + BNC: nước/(water) rất **cứng**/(hard) là/(is) nước/(water) cao/(high) hàm\_lượng/(content) khoáng\_sản/(mineral) trái/(in contrast) với/(with) mềm/(soft) ra nước/(water).

Clearly, ambiguous words in above example were translated precisely in the target language when utilizing WSD and BNC. In the first example, the word *hard* in *hard water* is translated to *cứng* (a type of water) which is more accurate than *chăm chi* (a personality) and *khó* (a difficulty).

## 4.3 The impact of context on WSD and WSD on SMT system

In many cases, the evaluation result of WSD is incorrect, resulting in the effect on the translation outcome of SMT. Below are two main reasons for this phenomenon: First, after the BNC expansion, the context could not embrace all possible cases due to limitation of contexts of BNC. Second, in several situations, information contexts of surrounding sentences should be used to determine labels of ambiguous words, whereas the system only uses the information in one sentence.

Besides, in the integration of WSD system into SMT system, WSD system occupies only a certain weight thus translation results are depend majorly on other models such as language model, translation model even though WSD gave precise results.

## 5 Conclusions

In this paper, we indicated a considerable effect of WSD which is bootstrapped on SMT system. The analyses and results on experiments point out that the approach of enhancing quality of WSD model contributes to the improvement of translation quality. The explanation for the increase of BLEU point is the impact of sparse data on the training set in WSD model. The expansion of training data from BNC whereby not only increases the degree of accurateness of WSD system but also improves the quality of translation. In the future, we would like to continue to experiment with the expansion of the training set on other sources to enhance the quality of translation.

## Acknowledgments

This paper has been supported by VNU project "Exploiting Very Large Monolingual Corpora for Statistical Machine Translation" (code QG.12.49).

## References

- Vamshi Ambati, Stephan Vogen and Jaime Carbonell. 2011. *Multi-Strategy Approaches to Active Learning for Statistical Machine Translation*. Proc of the 13th Machine Translation Summit.
- Marine Carpuat and Dekai Wu. 2007. *Improving Statistical Machine Translation Using Word Sense Disambiguation*. Proceedings of EMNLP-CoNLL.
- Y. S. Chan, H. T. Ng, and D. Chiang. 2007. *Word Sense Disambiguation Improves Statistical Machine Translation*. Proceedings of ACL.
- Jeremy H. Clear 1993. *The British National Corpus* MIT Press, Cambridge, MA, USA, pages 163–187.
- Philipp Koehn, Franz Josef Och, Daniel Marcu. 2003. *Statistical Phrase-based Translation*. In Proceedings of HLT-NAACL.
- Philipp Koehn et al. June, 2007. *Moses: Open Source Toolkit for Statistical Machine Translation*. In ACL, demonstration session, Prague, Czech Republic.
- C.A. Le and A. Shimazu. 2004. *High Word Sense Disambiguation Using Naive Bayesian Classifier with Rich Features*. The 18th Pacific Asian Conference on Linguistic Information and Computation (PACLIC18), pages 105–113.
- Nguyen, T. P., Nguyen V. V. and Le A. C. 2003. *Vietnamese Word Segmentation Using Hidden Markov Model*. In Proceedings of International Workshop for Computer, Information, and Communication Technologies in Korea and Vietnam.
- Franz Josef Och and Hermann Ney. 2000. *Improved Statistical Alignment Models*. In Proceedings of ACL.
- Pham, N. H., Nguyen L. M., Le A. C., Nguyen P. T., and Nguyen V. V. 2003. *LVT: An English-Vietnamese Machine Translation System*. In Proceedings of FAIR.
- Stolcke, A. September, 2002. *SRILM - An Extensible Language Modeling Toolkit*. In Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado.
- Varea, I. G., F. J. Och, H. Ney, and F. Casacuberta. 2001. *Refined Lexicon Models for Statistical Machine Translation using a Maximum Entropy Approach*. Proceedings of ACL, pages 204–211.
- Bird, Steven, Ewan Klein and Edward Loper. 2006. *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media, 2009
- Thomas M. Cover and Joy A. Thomas: *Elements of Information Theory*. New Jersey, John Wiley & Son.
- Och F.J. 2003 *Minimum Error Rate Training in Statistical Machine Translation*. Proceedings of the 41st International Conference on Computational Linguistics, pages 160–167.

# Orthographic and Morphological Processing for Persian-to-English Statistical Machine Translation

Mohammad Sadegh Rasooli and Ahmed El Kholly and Nizar Habash

Center for Computational Learning Systems

Columbia University, New York, NY

{rasooli, akholly, habash}@cccls.columbia.edu

## Abstract

In statistical machine translation, data sparsity is a challenging problem especially for languages with rich morphology and inconsistent orthography, such as Persian. We show that orthographic preprocessing and morphological segmentation of Persian verbs in particular improves the translation quality of Persian-English by 1.9 BLEU points on a blind test set.

## 1 Introduction

In the context of statistical machine translation (SMT), the severity of the data sparsity problem, typically a result of limited parallel data, increases for languages with rich morphology such as Arabic, Czech and Turkish. The most common solution, other than increasing the amount of parallel data, is to develop language-specific preprocessing and tokenization schemes that reduce the overall vocabulary and increase the symmetry between source and target languages (Nießen and Ney, 2004; Lee, 2004; Oflazer and Durgar El-Kahlout, 2007; Stymne, 2012; Singh and Habash, 2012; Habash and Sadat, 2012; El Kholly and Habash, 2012). In this paper, we work with Persian, a morphologically rich language with limited parallel data. Furthermore, Persian’s standard orthography makes use of a combination of spaces and semi-spaces (zero-width non-joiners), which are often ignored or confused, leading to orthographic inconsistencies and added sparsity. We address the orthographic challenge of inconsistent spacing with a supervised learning method which successfully recovers near all spacing errors. We also present a set of experiments for morphological segmentation to help improve Persian-to-English SMT. We show that the combination of orthographic cleanup and morphological segmentation for verbs in particular improves over a simple preprocessing baseline.

## 2 Related Work

Much work has been done to address data sparsity in SMT employing a variety of methods such as morphological and orthographic processing (Nießen and Ney, 2004; Lee, 2004; Goldwater and McClosky, 2005; Oflazer and Durgar El-Kahlout, 2007; Stymne, 2012; Singh and Habash, 2012; Habash and Sadat, 2012; El Kholly and Habash, 2012), targeting specific out-of-vocabulary phenomena with name transliteration or spelling expansion (Habash, 2008; Hermjakob et al., 2008) or using comparable corpora (Prochasson and Fung, 2011). Our approach falls in the class of orthographic and morphological preprocessing.

Previous research on Persian SMT is rather limited despite some early efforts (Amtrup et al., 2000). A few parallel corpora have been released, such as (Pilevar et al., 2011; Farajian, 2011). We conduct our research on an unreleased Persian-English parallel corpus (El Kholly et al., 2013a; El Kholly et al., 2013b).

In terms of preprocessing efforts, Kathol and Zheng (2008) use unsupervised Persian morpheme segmentation. Other attempts to improve Persian SMT use syntactic reordering (Gupta et al., 2012; Matusov and Köprü, 2010) and rule-based post editing (Mohaghegh et al., 2012). El Kholly et al. (2013a) and El Kholly et al. (2013b) also address resource limitation for Persian-Arabic SMT by pivoting on English.

Our approach is similar to Kathol and Zheng (2008), except that we do not use unsupervised learning methods for segmenting morphemes and we explore POS-specific processing instead of segmenting all words. We make extensive use of available resources for Persian morphology such as the Persian dependency treebank (Rasooli et al., 2013), the Persian verb analyzer tool (Rasooli et al., 2011a), the Persian verb valency lexicon (Rasooli et al., 2011c), and the PerStem Persian segmenter (Jadidinejad et al., 2010).

### 3 Persian Orthography and Morphology

#### 3.1 Orthography

Persian is written with the Perso-Arabic script. Unlike Arabic, some Persian words have inter-word zero-width non-joiner spaces (or semi-spaces). Many writers incorrectly write the semi-spaces as regular spaces (Shamsfard et al., 2010). This causes data inconsistency and some word-sense ambiguity, e.g., if the word نام آشنا<sup>1</sup> *nAm\_ĀšnA*<sup>1</sup> ‘reputed’ (adjective) is written with regular spaces, its meaning becomes ‘the familiar name’. While humans may be able to recover, typical natural language processing tools will fail since they expect standard Persian spelling.

#### 3.2 Morphology

Persian has a heavily suffixing affixational morphology with no expression of grammatical gender (Amtrup et al., 2000). We give a brief description of Persian adjectives, nouns and verbs and compare to English.

**Adjectives** Persian adjectives have a limited inflection space: they may be simple, comparative or superlative. In comparative and superlative forms (except for Arabic loan words), a suffix attaches to the adjective: *+tar*<sup>2</sup> ‘+er’ for comparative and *+tarīn* ‘+est’ for superlative adjectives. English uses both suffixes (‘+er/+est’) and multi-word construction with ‘more/most’, in addition to some irregular cases such as ‘good’, ‘better’, and ‘best’. As such, it might be hard to define a consistent preprocessing scheme for adjectives in Persian with respect to English.

**Nouns** Nouns are generally similar to English. For example, like English, a suffix marks plural number: mostly *+ha* and sometimes *+ān*. Exceptions include Arabic broken plural loan words. Unlike English, Persian has a suffixing indefinite marker (*+i*) comparable in meaning to English’s ‘a’ or ‘an’ indefinite particles. In Persian noun phrases consisting of a noun followed by one or more adjectives, the indefinite suffix attaches to the last adjective.

**Verbs** A verb in Persian may be inflected in different combinations for tense, mood, aspect, voice and person. There are many interesting

phenomena in Persian verbs, e.g. the past tense stem is used with another auxiliary verb to create the future form. When an auxiliary verb is used, prefixes attach to the auxiliary verb instead of the root. The negative marker (*+n*) ‘not’ and the object pronouns are attached to the verbs, leading to more than 100 verb conjugated forms (Rasooli et al., 2011b). For example, the verb *نمی خواندمش* *nmy\_xwAndmš* can be tokenized to *n+ my+ xwAnd +m +š* ‘I was not reading it’ [lit. ‘not+ was(continuous)+ read(past)+I+it’]. Persian is a pro-drop language; almost half of the verbs in the Persian dependency treebank do not have an explicit subject (Rasooli et al., 2013). By comparison, English has a much simpler verbal morphology with explicit subject realization. This suggests that tokenizing Persian verbs may be helpful to Persian-English SMT in that it reduces sparsity and increases symmetry with English.

### 4 Space Correction

In standard Persian orthography, semi-space characters show inter-word boundaries. Around 8% of all tokens in Persian dependency treebank have semi-spaces (Rasooli et al., 2013). However, in real Persian text, many of these semi-spaces are written as regular space. Although semi-space restoration may actually increase sparsity by creating more compounded forms of words, it is an important step to allow the use of Persian morphological resources that expect their presence.

In order to improve the quality of spacing in Persian texts, we use a language-modeling approach to correct spacing errors. The approach relies on the existence of a lexicon of semi-spaced words. The lexicon provides a mapping model from the regular-spaced versions of the words to their correct semi-spaced version. Starting with a sentence, we identify all sequences of regularly spaced words that can be mapped to semi-spaced versions. An expanded lattice version of the sentence including both forms is then decoded with a language model to select the path with the highest probability.

In terms of resources, we use the Peykare corpus (Bijankhan et al., 2011) and Persian dependency treebank (Rasooli et al., 2013) to create the semi-space lexicon and language model. The training data consists of about 398 thousand sentences and 89 million tokens (12 million types). To construct the lexicon, we extract all words with semi-spaces in the training data. We further extend the lexicon to cover known semi-space inflections for seen words, such as plural suffixes in nouns,

<sup>1</sup>We use the Habash-Soudi-Buckwalter Arabic transliteration (Habash et al., 2007) in the figures with extensions for Persian as suggested by Habash (2010). We show semi-spaces with underscore character.

<sup>2</sup>Suffixes that require a semi-space are marked in the transliteration with an underscore.

superlative and comparative suffixes in adjectives and prefixing continuous markers in verbs. The language model is a trigram model with back-off.

We use the development part of the Persian dependency treebank for tuning the n-gram model. On the test part of the Persian dependency treebank, we replace every semi-space with regular space and try to predict the semi-spaces with our model. The baseline accuracy (of having no semi-spaces) on the test set is 92.2%. Our system’s accuracy is 99.43%. The precision, recall and F-score of producing semi-spaces are 93.11%, 99.98% and 96.42%, respectively. The recall of our approach is almost perfect, but the precision is not as good, suggesting that we over assign semi-space. There are two common errors in the results. The first problem is with the hard distinction between adjectives and verbs, e.g., خراب شده *xrAb šdh* ‘dilapidated’ vs. خراب شده *xrAb šdh* ‘has destroyed’. The second problem is with errors in the training data, especially from the Peykare corpus (Bijankhan et al., 2011).<sup>3</sup>

## 5 Morpheme Segmentation

In this section, we present the two different morphological segmentation methods: PerStem and VerbStem.

**PerStem** As a baseline method for morphological segmentation, we use the off-the-shelf Persian segmenter, PerStem (Jadidinejad et al., 2010).<sup>4</sup> PerStem is a deterministic tool employing a set of regular expressions and rules for segmenting Persian words. PerStem separates most affixes for all parts-of-speech when applicable. PerStem has been used by other researchers for tokenization purposes (El Kholy et al., 2013a; El Kholy et al., 2013b).

**VerbStem** As discussed in Section 3, Persian verbs are particularly problematic for Persian-English SMT because of their rich morphology and differences from English. We experiment with targeting Persian verbs for segmentation. To identify which words are verbs, we use a simple maximum likelihood POS tagging model built on the

<sup>3</sup>Peykare is not actually written with semi-spaces. However, each word unit (consisting of one or more tokens) is written on one line and it is almost straightforward to standardize the corpus and add the semi-spaces. Unfortunately, some word lines in this corpus have two or more words that should have been written on separate lines, which leads to false examples of inserted semi-spaces, e.g., هنگامی که *hngAmy\_kh* ‘when that’ should be written with regular space instead of semi-space.

<sup>4</sup><http://sourceforge.net/projects/perstem/>

Peykare corpus (Bijankhan et al., 2011). For analysis and segmentation, we use an available Persian verb analyzer tool (Rasooli et al., 2011a)<sup>5</sup> and extend it with a deterministic segmentation algorithm to allow us to generate the needed tokens.<sup>6</sup> For each verb, we segment the negative marker, continuous marker, subject pronoun, object pronoun, participle marker, and prefix marker from the verb stem. We add spaces to the end of prefixes and beginning of suffixes, e.g., نمی خواندمش *nmy\_xwAndmš* would be segmented into *n my\_xwAnd m š*.<sup>7</sup> In our segmentation scheme, we do not perform any reordering nor try to address compound verbs in Persian.

Both the POS model and the Persian verb analyzer/segmenter expect the input text to have standard semi-space usage. Thus, we have to apply this step after semi-space correction. Figure 1 presents an example in different representations.

## 6 MT Evaluation

**Experimental Settings** We conduct several experiments using different segmentation decisions: **Raw** is original text; **Raw-RS** is Raw text but with regular spaces replacing all semi-spaces; **PerStem** is text processed with PerStem; **Clean-SS** is text with automatically corrected semi-spaces; and **VerbStem** is text processed with the verb segmentation method discussed in the previous section. Figure 1 compares three versions of the same sentence processed in different methods.

We use a Persian-English parallel corpus consisting of about 160 thousand sentences and 3.7 million words for translation model training (El Kholy et al., 2013a; El Kholy et al., 2013b). Word alignment is done using GIZA++ (Och and Ney, 2003). For language modeling, we use the English Gigaword corpus with 5-gram LM implemented with the KenLM toolkit (Heafield, 2011). All experiments are conducted using the Moses phrase-based SMT system (Koehn et al., 2007) with a maximum phrase length of 8. The decoding weight optimization uses a set of 1,000 sentences extracted randomly from the parallel corpus. We use only one English reference for tuning. We report results on a dev set and a blind test set, both with 268 sentences and three English references.

<sup>5</sup><https://github.com/rasoolims/PersianVerbAnalyzer>

<sup>6</sup>We also update the verb list in the Persian verb analyzer using the Persian verb valency lexicon [version 3.0.1] (Rasooli et al., 2011c).

<sup>7</sup>We considered adding plus sign to the end of prefixes and beginning of suffixes, but this representation did worse in SMT experiments.

Input	از فردا نمی ترسم چرا که دیروز را دیده ام و امروز را دوست دارم
Raw-RS	Az frdA <b>nmy trsm</b> crAkh dyrwz rA <u>dydh</u> Am w Amrwz rA dwst dAr m from tomorrow , it would not have seen am yesterday and today i love
PerStem	Az frdA <b>nmy trsm</b> crAkh dyrwz rA <u>dy dh</u> Am w Amrwz rA dwst dAr m from tomorrow , am not seen since yesterday and today i love
VerbStem	Az frdA <b>n my trs m</b> crAkh dyrwz rA <u>dyd h</u> Am w Amrwz rA dwst dAr m from tomorrow , not afraid because i have seen yesterday and today i love
Reference	<b>i 'm not afraid</b> of tomorrow because <i>i have seen yesterday</i> and i like today

Figure 1: Example output from three systems and one of the references from the dev set. As seen in the bolded and underlined words, the VerbStem system captures linguistic information and produces better translation quality.

Method	Raw	Raw-RS	PerStem	Clean-SS	VerbStem
BLEU	33.0	33.6	32.6	32.2	<b>33.7</b>

Table 1: SMT results on the dev set.

Model	BLEU	METEOR	TER
Raw-RS (Baseline)	31.4	31.2	<b>60.9</b>
VerbStem (Best model)	<b>33.3</b>	<b>32.2</b>	61.1

Table 2: Results from the baseline and the best system on the blind test set.

**Results and Discussion** The results of SMT experiments on the dev set are shown in Table 1. VerbStem is our best system. Simply replacing all spaces (Raw-RS) does rather well and is plausibly the strongest simplest baseline we can compare to. PerStem and Clean-SS underperform the baseline. Clean-SS is the worst system (as expected since it increases sparsity), but it is necessary as a step for VerbStem. The improvement in VerbStem is possibly the result of reduced sparsity and increased symmetry between English and Persian. Verb segmentation makes a lot of information explicit, such as negation, subject pronoun (especially since Persian as a pro-drop language) and object pronoun.

We apply VerbStem to the blind test set and compare it to Raw-RS. Table 2 shows the blind test results using BLEU-4 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007) and TER (Snover et al., 2006). VerbStem produces a higher BLEU score improvement over the Raw-RS baseline on the blind test compared to the dev set. This may suggest that our dev set is easier in general. Although our best system does well in Figure 1, the best result still suffers from suboptimal word order. The position of the verb in Persian (as an SOV language) is very problematic when translating to English (an SVO language) especially for long sentences.

## 7 Conclusion and Future Directions

Our experiments show that segmenting Persian verbs improves translation quality. However, the translation output of all current systems in this paper suffer from word order problems. In the future, we plan to investigate how to improve word order in the translation output using a variety of techniques such as hierarchical phrase-based models (Chiang, 2005; Kathol and Zheng, 2008; Cohn and Haffari, 2013), or models employing parsers to be developed using the Persian dependency treebank (Collins et al., 2005; Elming and Habash, 2009; Carpuat et al., 2010).

**Acknowledgments** The second author was funded by a research grant from the Science Applications International Corporation (SAIC). We thank Nadi Tomeh for helpful discussions.

## References

- Jan Willers Amtrup, Hamid Mansouri Rad, Karine Megerdooian, and Rémi Zajac. 2000. *Persian-English machine translation: An overview of the Shiraz project*. Computing Research Laboratory, New Mexico State University.
- Mahmood Bijankhan, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. 2011. Lessons from building a Persian written corpus: Peykare. *Language Resources and Evaluation*.
- Marine Carpuat, Yuval Marton, and Nizar Habash. 2010. Improving Arabic-to-English Statistical Machine Translation by Reordering Post-Verbal Subjects for Alignment. In *ACL'10*.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL'05*.
- Trevor Cohn and Gholamreza Haffari. 2013. An infinite hierarchical bayesian model of phrasal translation. In *ACL'13*.
- Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *ACL'05*.

- Ahmed El Kholly and Nizar Habash. 2012. Orthographic and morphological processing for English–Arabic statistical machine translation. *Machine Translation*.
- Ahmed El Kholly, Nizar Habash, Gregor Leusch, Evgeny Matusov, and Hassan Sawaf. 2013a. Language independent connectivity strength features for phrase pivot statistical machine translation. In *ACL'13*.
- Ahmed El Kholly, Nizar Habash, Gregor Leusch, Evgeny Matusov, and Hassan Sawaf. 2013b. Selective combination of pivot and direct statistical machine translation models. In *IJCNLP'13*.
- Jakob Elming and Nizar Habash. 2009. Syntactic Reordering for English–Arabic Phrase–Based Machine Translation. In *EACL'09 Workshop on Computational Approaches to Semitic Languages*.
- Mohammad Amin Farajian. 2011. Pen: Parallel English–Persian news corpus. In *WORLD-COMP'11*.
- Sharon Goldwater and David McClosky. 2005. Improving statistical mt through morphological analysis. In *EMNLP'05*.
- Rohit Gupta, Raj Nath Patel, and Ritnesh Shah. 2012. Learning improved reordering models for Urdu, Farsi and Italian using SMT. In *Workshop on Reordering for Statistical Machine Translation*.
- Nizar Habash and Fatiha Sadat. 2012. Arabic preprocessing for statistical machine translation. *Challenges for Arabic Machine Translation*.
- Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.
- Nizar Habash. 2008. Four Techniques for Online Handling of Out-of-Vocabulary Words in Arabic–English Statistical Machine Translation. In *ACL'08*.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *WMT'11*.
- Ulf Hermjakob, Kevin Knight, and Hal Daumé III. 2008. Name translation in statistical machine translation: Learning when to transliterate. *ACL'08*.
- Amir Hossein Jadidinejad, Fariborz Mahmoudi, and Jon Dehdari. 2010. Evaluation of PerStem: a simple and efficient stemming algorithm for Persian. In *Multilingual Information Access Evaluation*.
- Andreas Kathol and Jing Zheng. 2008. Strategies for building a Farsi–English smt system from limited resources. In *INTERSPEECH'08*.
- 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 *ACL'07*.
- Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *WMT'07*.
- Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *NAACL'04*.
- Evgeny Matusov and Selçuk Köprü. 2010. Improving reordering in statistical machine translation from farsi. In *AMTA'10*.
- Mahsa Mohaghegh, Abdolhossein Sarrafzadeh, and Mehdi Mohammadi. 2012. GRAFIX: Automated rule-based post editing system to improve English–Persian SMT output. In *COLING'12*.
- Sonja Nießen and Hermann Ney. 2004. Statistical Machine Translation with Scarce Resources using Morpho-syntactic Information. *Computational Linguistics*.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*.
- Kemal Oflazer and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in English-to-Turkish statistical machine translation. In *WMT'07*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL'02*.
- Mohammad Taher Pilevar, Hesham Faili, and Abdol Hamid Pilevar. 2011. Tep: Tehran english–persian parallel corpus. In *CICLING'11*.
- Emmanuel Prochasson and Pascale Fung. 2011. Rare word translation extraction from aligned comparable documents. In *ACL'11*.
- Mohammad Sadegh Rasooli, Hesham Faili, and Behrouz Minaei-Bidgoli. 2011a. Unsupervised identification of Persian compound verbs. In *MICAI'11*.
- Mohammad Sadegh Rasooli, Omid Kashefi, and Behrouz Minaei-Bidgoli. 2011b. Effect of adaptive spell checking in Persian. In *NLPKE'11*.
- Mohammad Sadegh Rasooli, Amirsaied Moloodi, Manouchehr Kouhestani, and Behrouz Minaei-Bidgoli. 2011c. A syntactic valency lexicon for Persian verbs: The first steps towards Persian dependency treebank. In *LTC'11*.
- Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaied Moloodi. 2013. Development of a Persian syntactic dependency treebank. In *NAACL'13*.
- Mehrnoush Shamsfard, Hoda Sadat Jafari, and Mahdi Ilbeygi. 2010. Step-1: A set of fundamental tools for Persian text processing. In *LREC'10*.
- Nimesh Singh and Nizar Habash. 2012. Hebrew morphological preprocessing for statistical machine translation. In *EAMT'12*.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA'06*.
- Sara Stymne. 2012. *Text Harmonization Strategies for Phrase-Based Statistical Machine Translation*. Ph.D. thesis, Linköping.



# Interoperability between Service Composition and Processing Pipeline: Case Study on the Language Grid and UIMA

Mai Xuan Trang, Yohei Murakami, Donghui Lin, and Toru Ishida

Department of Social Informatics, Kyoto University  
Yoshida-Honmachi, Sakyo-Ku, Kyoto, 606-8501, Japan  
trangmx@ai.soc.i.kyoto-u.ac.jp, {yohei, lindh,  
ishida}@i.kyoto-u.ac.jp

## Abstract

Integrating language resources is a critical part in building natural language processing applications. Processing pipeline and service composition are two approaches for sharing and combining language resources. However, each approach has its drawback. While the former lacks consideration about property rights of language resources, the later is not efficient to process and transfer huge amount of data through web services. In this paper we address the issue of interoperability between two approaches to mutually complement their disadvantages. We show an integration of service composition and processing pipeline, and how the integration can be used to help developers seamlessly build NLP applications. We then present a case study that adopts the integration to integrate two representative frameworks: the Language Grid and UIMA.

## 1 Introduction

The creation of language resources (LRs) remains a fundamental activity in the field of language technology. The number of language resources has been increasing year by year. Based on these resources, developers build advanced Natural Language Processing (NLP) applications (hereafter referred to as the applications) such as Watson and Siri by combining some of these resources. However, it is difficult for developers to collect and combine the most suitable set of language resources in order to achieve the developers' goals.

There are two types of language resource coordination frameworks supporting developers sharing and combining language resources and tools: Framework-based processing pipeline such as GATE (Cunningham et al. 2002) and UIMA (Ferrucci et al., 2004) and framework-based service

composition such as the Language Grid (Ishida, 2006). Interoperability between components in one framework is dealt by defining Common Data Exchange or standard interface for components. For example, UIMA defines Common Analysis Structure as data exchange between components, the Language Grid defines standard interfaces in a ontology for their language services (Hayashi, 2007). Interoperability among formats of two processing pipeline frameworks UIMA and GATE is explored in (Ide et al., 2009a). This paper addresses the issue of how to bridge the gap between two data structures of common data exchange format. In this work we focus on interoperability of two different types of frameworks.

Therefore, this paper realizes interoperability between those two types of frameworks to mutually complement their disadvantages. To this end, we address the following issues:

- Integration between two types of frameworks: Service composition and processing pipeline. The integration provides ability to wrap components of one framework as components of another. This will lead to more language resources and tools becoming available in both frameworks, facilitating the development process of NLP applications.
- A case study of integration two representative frameworks: The Language Grid and UIMA is implemented to realize the integration concept framework.

The remainder of this paper is organized as follows: in section 2 we will briefly discuss features of the two types of language resource coordination frameworks. The integration of the service composition and processing pipeline will be presented in section 3. We show a case study on integration between the Language Grid and UIMA in section 4. Finally, section 5 concludes this paper.

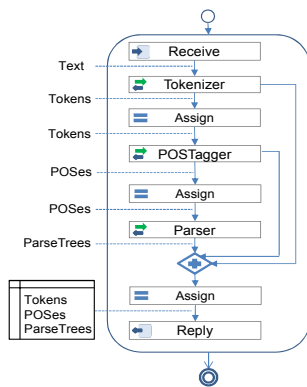


Figure 1: Service composition approach

## 2 Language Resource Coordination Frameworks

### 2.1 Service Composition Approach

In this approach, language resources are wrapped as web services that users can combine to create customized composite language services for their need. Figure 1 shows a composite service composing three language services: Tokenizer, POSTagger and Parser. Each service in the workflow is defined by an interface with input and output. Interoperability between services in a workflow is ensured by conforming interfaces of the services. Output of a previous service and input of the later service in the workflow must be compatible.

Language resources available on these frameworks are provided by variety of providers. For instance, PANACEA currently has more than 160 services provided by 11 service providers. On the Language Grid, over 170 services are provided by 140 groups from 17 countries. Providers need to protect their resources with intellectual rights, so that they can configure permission and monitor usage statistics of their resources. Service composition approach provides access control functionality to deal with this issue. This advantage encourages providers to share their language resources, increasing availability of language services.

### 2.2 Processing Pipeline Approach

This approach focuses on providing a setting for creating analysis pipelines, oriented towards linguistic analysis and stand-off annotation model. The purpose of these frameworks is to combine language resources to analyze huge amounts of data at the local environment.

Processing tools are combined into a pipeline to analyze documents. Each tool is defined as an annotator to annotate the document with anno-

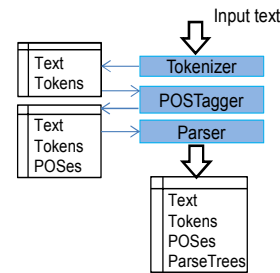


Figure 2: Processing pipeline approach

tations represented as stand-off annotation. The document together with annotations is formed in a Common Data Exchange Format (CDEF). The CDEF document is then exchanged between components in the pipeline. Figure 2 shows a pipeline of three annotators: Tokenizer, POSTagger, and Parser. The pipeline enriches input text with three annotation types: Token, POS, and ParseTree.

A disadvantage of this approach is the lack of access control to share language resources distributedly with intellectual rights. This limits the availability of language resources.

## 3 Integration of Service Composition and Processing Pipeline

### 3.1 Mapping Service Interface Invocation and Stand-off Annotation

The CDEF data structure is defined based on widely used de-facto standards such as TEI (Vanhoutte, 2004), CES (Ide, 2000), and *common interface format* being developed under the context of ISO committee TC 37/SC 4 (Ide, 2009b). CDEF basically consists of two parts: one representing document text, and the other representing annotations. Figure 3 shows an example of CDEF in XML-based format:

- `<doc>`: represents the document, the `id` attribute is used to distinguish documents when a pipeline processing with multiple documents.
- `<annotations>`: represents all annotations produced by a pipeline. An annotation is described by `<annot>` tag, the `type` attribute indicates type of the annotation, two attributes `begin` and `end` define annotation's offset and the `componentID` attribute shows the annotator producing this annotation. The structure of the annotation is defined by feature structure (`fs`) tag and feature (`f`) tags.

Each language service has its own interface with input and output. For an annotator in processing pipeline, we can assume that it's input

```

<?xml version="1.0" encoding="UTF-8"?>
<annotatedDoc>
  <doc id="1" mimeType="text"
    docString="Text of the document"/>
  <annotations>
    <annot type="POS" docID="1" begin="1"
      end="5" componentID="POSTager">
      <fs>
        <f name="lemma" value="Text"/>
        <f name="postag" value="noun"/>
        ...
      </fs>
    </annot>
    ...
  </annotations>
</annotatedDoc>

```

Figure 3: Structure of common data exchange

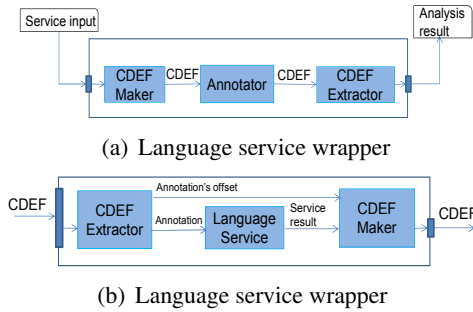


Figure 4: Wrappers

and output are CDEF. The mapping is defined to map input/output of language services with annotation types in CDEF. We define CDEF Maker and CDEF Extractor to conduct the mapping and create two wrappers: Language Service Wrapper and Annotator Wrapper as shown in Figure 4(a) and Figure 4(b) respectively. The former is used to wrap an annotator as a language service, the later is used to wrap a language service as an annotator:

- CDEF Extractor manipulates with CDEF to extract annotation and maps it with input/output of a language service. The Extractor uses XML parsing technique such as DOM and SAX to parse CDEF document and extract annotation. the annotation type and offset are extracted from the element `<annot>`. The annotation structure with features and values is extracted from `<fs>` node and sub-nodes `<f>`s. The Extractor then maps the annotation with a corresponding language service type which is served as input or output of a language service.
- CDEF Maker maps input/output of language services to annotation types and creates CDEF document. When wrapping an annotator as a language service with defined input and output, CDEF Maker first finds the offset of the defined input in the original text and then maps it with an annotation. Finally, it creates CDEF document from the original text and the annotation. In case of the input is

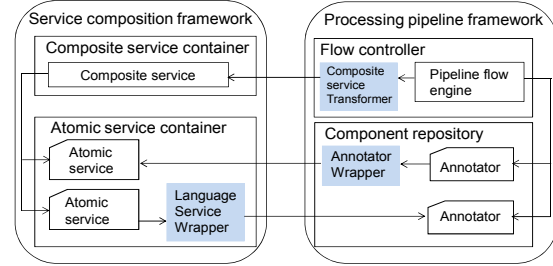


Figure 5: Integration Framework

text, the CDEF document is created with only *doc* part. When wrapping a language service as an annotator, CDEF Maker maps structure of the language service output with structure of a corresponding annotation type and use annotation's offset, extracted by CDEF Extractor, to create an annotation. This annotation is then added to the CDEF document.

### 3.2 Integration Framework

Integration framework enables users easily combine both types of components: language service and annotator. Users can use annotators to create composite services, use language services in a pipeline flow, or use both in composite services or in pipeline flows. It also provides ability to create a pipeline flow from a composite service.

Figure 5 illustrates integration of service composition and processing pipeline. A wrapper system consisting of Language Service Wrapper and Annotator Wrapper is used in this integration framework. Annotator providers use the Language Service Wrapper to wrap an annotator as a language service. This service is then shared with access control in the service composition framework. Users who have access rights to the system can invoke this service or can use this service to compose composite services. The language service providers use the Annotator Wrapper to wrap a language service into an annotator, this annotator can be executed and combined in a pipeline flow.

Language resources are shared as language services with intellectual rights, it is easy to create composite services. However, pipeline flow has better performance when processing large amounts of data compared to composite service. We define Composite Service Transformer to transfer composite services into pipeline flows. A composite service contains information about binding services. From the binding services, names, providers and sequence of language resources using in the composite service can be ex-

tracted. The transformer uses this information to build an abstract pipeline flow of these language resources. Later on, developers will negotiate with the providers to get the concrete language resources for the pipeline.

#### 4 Case study: Integration of the Language Grid and UIMA

Using the integration framework concept, we integrate the Language Grid and UIMA. We implement two wrappers: Language Service Wrapper and Analysis Engine Wrapper. The former is used to wrap an analysis engine as a language service, while the later is used to wrap language service into an Analysis Engine. A composite service transformer is also implemented to help developers transfer composite services to UIMA flows.

CAS is common data exchanged between UIMA components. We implement CAS Maker and CAS Extractor to manipulate with CAS document and create the wrappers:

- CAS Extractor extracts annotation from CAS document and maps with input/output types of language services.
- CAS Maker maps the input/output types of language services with UIMA annotation types and creates CAS documents which are served as input/output of an analysis engine.

We use some libraries from the Language Grid and UIMA such as *jp.go.nict.langrid.client.ws\_1.2.\** and *org.apache.uima.\** to manipulate with the Language Grid types and UIMA CAS. We also defined a new language service interface in the Language Grid to represent an analysis engine. This service interface has *analyze* operation with input is a string representing document, and output is a collection of annotations.

A mapping between UIMA annotation types and the Language Grid types is defined. We collect popular UIMA types defined for popular NLP functionalities. For each UIMA type we find a corresponding type in the Language Grid and create a mapping between these two types. For example, a *uima.annotation.Lemma* annotation can be mapped with *langrid.types.Morphem* type in the Language Grid, since these types contain similar morphological information such as *partOfSpeech*.

Composite Service Transformer extracts information about language resources used in a composite service and builds an UIMA flow by creating a descriptor file of the flow from the infor-

mation. This process may be complex, since it is transformation between two different types of flow. To facilitate the transformer, we adapt UIMA Flow Engine into the Language Grid Composite Service Container (Murakami et al. 2011), so that users can use this engine to create composite services. This kind of composite service is much easier to be transferred into an UIMA flow.

Analysis engines are wrapped as web services and shared in the Language Grid. Developers can easily collect and combine services to build a workflow for their application. However, using web services, transformation of huge data is not efficient. After testing the workflow with small amount of data and examine the output, if it satisfies the users requirement, then this workflow is transferred to a UIMA flow. Moreover, with the integration we can create hybrid applications combining analysis engines and language services.

The integration of UIMA and the Language Grid enhances the number of language resources available in both frameworks. Especially, this increases the number of language services related NLP in the Language Grid, and increases the robustness of the Language Grid.

#### 5 Conclusion

In this paper we proposed an integration of two types of language resource coordination frameworks: framework-based service composition and framework-based processing pipeline. The main contributions of this paper are as follows:

- The integration framework increases availability of language resources. Thus, it facilitates the process of creating applications.
- Integration of the Language Grid and UIMA is implemented to realize the framework.

In this paper, the type mapping between different frameworks is manually created. This technique is not very sufficient, due to the significant increase in number of types. Our future work will focus on using ontologies or extendable type system for a better approach of the type mapping.

#### Acknowledgments

This research was partly supported by a Grant-in-Aid for Scientific Research (S) (24220002, 2012-2016) from Japan Society for Promotion of Science (JSPS) and Service Science, Solutions and Foundation Integrated Research Program from JST RISTEX.

## References

- Bel N. 2010. Platform for Automatic, Normalized Annotation and Cos-Effective Acquisition of Language Resources for Human Language Technologies: PANACEA. In *Proceedings of the 26th Annual Congress of the Spanish Society for Natural Language Processing (SEPLN-2010)*.
- Cunningham H., Maynard D., Bontcheva K., and Tablan V. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.
- Ferrucci D., and Lally A. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10, pp. 327-348.
- Hayashi Y. 2007. Conceptual Framework of an Upper Ontology for Describing Linguistic Services. In: *Toru Ishida, Susan R. Fussel, Piek T. J. M. Vossen (Eds.): Intercultural Collaboration, LNCS 4568, Springer-Verlag, pp.31-45*.
- Ide, N., Bonhomme, P., and Romary, L. 2000. An XML-based Encoding Standard for Linguistic Corpora In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pp. 825-830.
- Ide N., and Suderman L. 2009a. Bridging the Gaps: Interoperability for GrAF, GATE and UIMA. In *Proceeding of the Third Linguistic Annotation Workshop. Singapore, August 2009*, pp. 27-34.
- Ide, N., and Romary, L. 2009b. Standards for language resources. *arXiv preprint arXiv:0909.2719*.
- Ishida T. 2006. An Infrastructure for intercultural collaboration. In *IEEE/IPSJ Symposium on Applications and the Internet (SAINT-06)*, pp. 96100.
- Kano Y., Miwa M., Chohen K. B., Hunter L. E., Ananiadou S., and Tshujii J. 2011. U-Compare: A modular NLP workflow construction and evaluation system. *IBM Journal of Research and Development*, 55(3).
- Murakami Y., Lin D., Tanaka M., Nakaguchi T., and Ishida T. 2011. Service Grid Architecture. In *The Language Grid: Service Oriented Collective Intelligence for Language Resource Interoperability*, Ishida T., Ed. Springer 2011, pp. 19-34..
- Schäfer U. 2006. Middleware for Creating and Combining Multi-dimensional NLP Markup. In *Proceedings of the EACL-2006 Workshop on Multi-Dimensional Markup in Natural Language Processing. Trento, Italy, April 2006*, pp. 8184.
- Vanhoutte, E. 2004. An Introduction to the TEI and the TEI Consortium. *Literary and linguistic computing*, 19(1), 9-16.

# Improving Calculation of Contextual Similarity for Constructing a Bilingual Dictionary via a Third Language

Takashi Tsunakawa    Yosuke Yamamoto    Hiroyuki Kaji  
Graduate School of Informatics, Shizuoka University  
3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka 432-8011, Japan  
{tuna, kaji}@inf.shizuoka.ac.jp

## Abstract

A novel method is proposed for measuring contextual similarity by “weighted overlapping ratio (WOR)” to construct a bilingual dictionary of a new language pair from two bilingual dictionaries sharing one language. The WOR alleviates the effect of a noisy seed dictionary resulting from merger of two bilingual dictionaries via a third language. Combined use of two word-association measures for extracting contexts from corpora is also proposed to compensate their weaknesses. Experiments on constructing a Japanese-Chinese dictionary via English show that the proposed method outperforms the conventional method based on cosine similarity.

## 1 Introduction

With the growth of communication via the Internet, people have more chance to access documents written in various languages. The number of Internet users in the Arabic, Russian, and Chinese languages have increased at least tenfold times in the last decade. It was sufficient in the past to bridge the gap between English and another language by using bilingual resources and services. For directly accessing Web contents written in various languages, however, multilingual dictionaries, translation, and information retrieval are required.

The present study focuses on a so-called triangulation approach for constructing a bilingual dictionary by merging two bilingual dictionaries sharing one language. For example, if Chinese-to-English and English-to-Arabic dictionaries are available, a Chinese-to-Arabic dictionary can be derived by collecting pairs of Chinese and Arabic terms (hereafter, “term pairs”) that have

common English translations. A serious problem with this approach, however, is how to filter out false term pairs, caused by polysemy of English terms. To validate each term pair as translations, we calculate the context similarity between the terms on the basis of the distributional hypothesis (Harris, 1954).

In triangulation approach, calculation of the context similarity in different languages is required. Previous studies have calculated context similarities by context vector projection onto another language by using a seed bilingual dictionary. Though this approach is effective, translation perplexity caused by context vector projection may cause negative effects described in Section 3.3. Instead of projection, our proposed method avoids this problem by using a “weighted overlapping ratio,” which directly maps words in context vectors in different languages.

## 2 Related work

Tanaka and Umemura (1994) proposed a triangulation method of constructing a bilingual dictionary. Their method has been augmented by using semantic classes (Bond et al., 2001) and parts of speech and cognates (Zhang et al., 2005).

Several methods of constructing a bilingual dictionary from contextual similarity have been proposed (Rapp, 1995; Kaji and Aizono, 1996; Tanaka and Iwasaki, 1996; Fung and Yee, 1998; Rapp, 1999; Sammer and Soderland, 2007). Most of them are based on context vector projection. Rapp (1999) replaced a word in the context vector with the translation first appeared in the dictionary, while Fung and Yee (1998) gave each translation a weight inversely proportional to the order of the translation in the dictionary. As another provision for translating context vectors,

mutual projection of context vectors was proposed (Fišer et al., 2011). Adapting a seed bilingual dictionary to the domain of comparable corpora has been proved to be effective (Kaji, 2005; Morin and Prochasson, 2011).

Other approaches (Déjean and Gaussier, 2002; Daille and Morin, 2005; Hazem et al., 2011) proposed methods based on identification of second-order affinities. Kaji et al. (2008) created a correlation matrix of context words versus translations. Vulić and Moens (2012) proposed a bilingual LDA model in which the term pairs are obtained on the basis of similar distributions of language-independent latent topics.

### 3 Proposed method

The proposed method is overviewed in Figure 1. Each step of the proposed method is described in the following subsections.

#### 3.1 Merging bilingual dictionaries

It is supposed that a bilingual dictionary between a source language  $S$  and a target language  $T$  can be constructed via a third language  $P$ . A bilingual dictionary,  $D_{L,L'}$ , between two languages,  $L$  and  $L'$ , can be defined as a set of term pairs  $\{(w_l, w_{l'})\} \subseteq L \times L'$ , where term  $w_l \in L$  can be translated as term  $w_{l'} \in L'$ .<sup>1</sup>

It is assumed that two bilingual dictionaries,  $D_{S,P}$  and  $D_{P,T}$ , are available. The merged bilingual dictionary between  $S$  and  $T$ , namely,  $D_{S,T}$ , is obtained from

$$\{(w_s, w_t) | \exists w_p: (w_s, w_p) \in D_{S,P} \wedge (w_p, w_t) \in D_{P,T}\}.$$

Note that term  $w_s$  cannot necessarily be translated into term  $w_t$  because of polysemy of term  $w_p$ . Such term pair  $(w_s, w_t)$  makes “noise” in the merged dictionary.

$D_{S,T}$  is used as a candidate set of term pairs to be ranked. It is also used as a seed bilingual dictionary to calculate the similarity of contexts in languages  $S$  and  $T$ .

#### 3.2 Extracting context

Spurious term pairs in the merged bilingual dictionary should be removed. The similarity of senses of each term pair is estimated by comparing their contexts. We represent the context of term  $w$  as a *weighted set of associated words*, i.e., words that are semantically or topically related with  $w$ . The weighted set of associated words,

<sup>1</sup> We describe that the vocabulary set of a language  $L$  also as  $L$  in short.

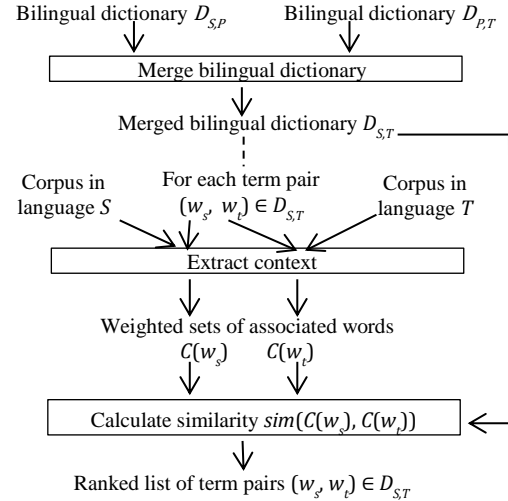


Figure 1. Overview of proposed method

$C(w)$ , is denoted as  $\{w_1/\alpha_1, w_2/\alpha_2, \dots, w_K/\alpha_K\}$  where  $w_k$  is an associated word of  $w$ , and  $\alpha_k$  is its weight assigned by the word-association measure based on their co-occurrence frequencies.

#### (1) Using a single word-association measure

Among words that co-occur with  $w$  in the corpus, only words that have association measure scores in the top- $M\%$ <sup>2</sup> are kept as associated words. The word-association measures employed are log-likelihood ratio (LLR), pointwise mutual information (MI), chi-squared score ( $\chi^2$ ), and discounted log-odds ratio (LOR) (Laroche and Langlais, 2010).

#### (2) Using combination of word-association measures

Each association measure has its own weakness in capturing word association. For example, LLR tends to overestimate frequent words, while MI tends to do infrequent ones. In general, infrequent associated words have less possibility to be matched when comparing two sets of associated words. A combination of these measures is expected to compensate for each weakness.

Associated words whose first association measure is in the top- $M_1\%$  and second is in the top- $M_2\%$  are kept. A weight corresponding to each associated word is given by the second measure. Two kinds of combinations are considered: LLR-MI and LLR-LOR, which respectively represent the first and second measures.

### 3.3 Calculating similarity between weighted sets of associated words

<sup>2</sup> A fixed threshold value of the association score was not used for keeping associated words because the proposed method obtained better results in our experiment.

A problematic case of context vector projection is illustrated in Figure 2. For calculating contextual similarity, such as a cosine, the context vectors<sup>3</sup> must be projected onto associated-word dimensions in the same language. In this approach, associated words are duplicated by translation perplexity. In this example, each word associated with the Japanese word “石油” sekiyu ‘petroleum’ has several possible English translations. It yields unnecessary Chinese associated words such as “力” li ‘power’ and “细胞” xibao ‘cell (in the biological sense),’ and then falsely decreases the cosine value because the norm of the projected vector increases.<sup>4</sup>

To avoid this problem, two sets of associated words are directly compared. For two weighted sets of associated words,  $C(w_s) = \{w_k/\alpha_k\}$  and  $C(w_t) = \{w'_l/\alpha'_l\}$ , a *weighted overlapping ratio* (WOR) is defined as:

$$\text{sim}(C(w_s), C(w_t)) = \frac{1}{2} \left\{ \frac{\sum_{k \in P} \alpha_k}{\sum_k \alpha_k} + \frac{\sum_{l \in Q} \alpha'_l}{\sum_l \alpha'_l} \right\}$$

where  $P = \{k | \exists w'_l: (w_k, w'_l) \in D_{S,T}\}$ ,  $Q = \{l | \exists w_k: (w'_l, w_k) \in D_{T,S}\}$ , and  $D_{S,T}$  and  $D_{T,S}$  are seed bilingual dictionaries between languages  $S$  and  $T$ . Term pairs  $(w_s, w_t)$  in the merged dictionary are ranked in order of their WORs. An example calculation of WOR is shown in Figure 3. The side effect from a noisy seed dictionary is considered to be moderated, since an unnecessary term is added only once per noisy term.

## 4 Experiments

Experiments on constructing a Japanese-Chinese bilingual dictionary via English as a third language were carried out. Note that this approach is applicable for any language combinations. We report three kinds of comparison: WOR vs. cosine similarity, word-association measures, and newspaper corpus vs. Wikipedia corpus.

Window size  $W$  for counting co-occurrence frequencies was fixed to 10. Five-fold cross validation was conducted for optimizing parameters for choosing associated words ( $M$ ;  $M_1$  and  $M_2$ ).<sup>5</sup>

<sup>3</sup> The weighted set of associated words is compatible to the context vector with dimensions of associated words in the vector space model.

<sup>4</sup> Rapp (1999) substituted each associated word to only a single translation. In that case, however, a noisy seed dictionary significantly decreases the probability that the translation is appropriate.

<sup>5</sup> The optimized values were as follows:  $M = 1.02$  (%) for LOR and  $M_1 = 13.5$ ,  $M_2 = 9.4$  (%) for MI-LLR.

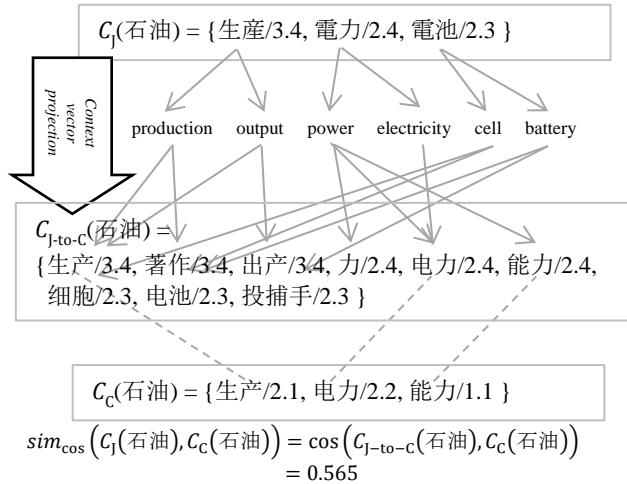


Figure 2. Calculation of contextual similarity by context-vector projection

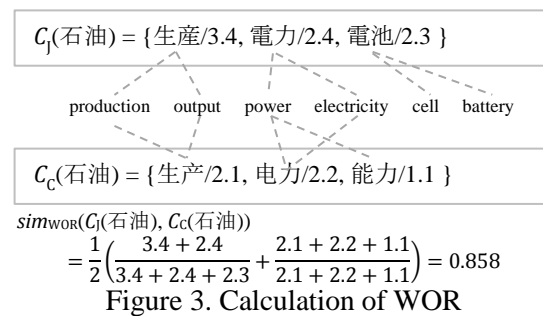


Figure 3. Calculation of WOR

The final evaluation score is output by averaging all five results.

### 4.1 Experimental settings

Two sets of comparable corpora were employed for our experiment: newspaper and Wikipedia. The newspaper set consists of the Mainichi Shimbun Corpus (2000-2010; 22.3GB) and the Xinhua News Corpus in LDC Chinese Gigaword (2000-2010; 4.24GB). The Wikipedia set consists of Japanese and Chinese Wikipedia articles dumped on September 2012 (Japanese: 821k articles; 3.1GB / Chinese: 520k articles; 0.7GB).

EDR Japanese-to-English/English-to-Japanese dictionaries and LDC Chinese-to-English dictionary<sup>6</sup> were used as input dictionaries. Each EDR dictionary has 376k word pairs, including 161k English distinct words and 221k Japanese distinct words. The LDC dictionary consists of 82k distinct word pairs.

3,000 term pairs (each term occurs at least 2,500 times in the corpus) were randomly extracted as the test data from the merged Japanese-Chinese dictionary. Each term pair was labeled as *translation* or *non-translation* by ma-

<sup>6</sup> The English-to-Chinese dictionary was obtained by inverting the LDC dictionary.



majority decision of three human annotators. The test set consists of 1,053 *translation* pairs and 1,947 *non-translation* pairs.

Precision  $P$  and recall  $R$  for the term pairs with the top- $\delta\%$  of WORs or cosine values were calculated. Some *translation* term pairs could not be correctly judged because those terms are sometimes only used for representing different senses from a sense in the comparable corpus. The recall therefore does not reach a higher value compared with the precision value. For this reason, an  $F_\beta$ -score with  $\beta = 0.5$  was adopted as the final evaluation score so as to emphasize precision as twice as important as recall. The best  $F_{0.5}$ -scores were obtained when  $\delta = 20$  (%) (see Table 1).

## 4.2 WOR vs. cosine similarity

To confirm the effect of WOR, it was compared with the conventional cosine by context vector projection. The best evaluation scores are listed in Table 2. WOR outperformed the cosine measure on both corpus sets.

The merged dictionary for comparing associated words can also be substituted by existing bilingual dictionaries between languages  $S$  and  $T$  if available. To examine the effect of using the noisy seed bilingual dictionary, additional experiments in using the EDR Japanese-Chinese dictionary (223k term pairs) as a seed dictionary were conducted. The  $F_{0.5}$ -score by WOR with this setting was 0.743, while 0.721 by cosine measure. The drop in the  $F_{0.5}$ -score by using the merged dictionary as the seed were 1.4 points by WOR, which were smaller than the drop (3.0 points) obtained by the cosine. This result shows that WOR is more robust than the cosine in the case that a noisy seed dictionary is used.

## 4.3 Single measure vs. combination of measures

Experiments on using all word association measures were carried out. Among the single word-association measure, the highest  $F_{0.5}$ -score of 0.689 was obtained by LOR as listed in Table 2, and it confirmed a previous comparative experiment (Laroche and Langlais, 2010). Both combinations of the multiple association measures outperformed LOR on the newspaper set. These results indicate that the weakness that LOR covered could also be covered by LLR.

## 4.4 Newspaper vs. Wikipedia as the comparable corpus

The characteristics of the results obtained from

$\delta$ (%)	$P$	$F_{0.5}$
10	0.908	0.611
20	0.833	<b>0.729</b>
30	0.744	0.723
40	0.640	0.659
50	0.567	0.605

Table 1. Evaluation scores attained some values of  $\delta$  (%) (settings: WOR, LLR-MI, newspaper)

	Corpus set	Newspaper		Wikipedia	
	Measure	$P$	$F_{0.5}$	$P$	$F_{0.5}$
WOR	LLR	0.721	0.631	0.664	0.646
	MI	0.704	0.616	0.711	0.691
	LOR	0.788	0.689	0.792	0.693
	$\chi^2$	0.717	0.622	0.717	0.632
	LLR-MI	<b>0.833</b>	<b>0.729</b>	<b>0.796</b>	<b>0.696</b>
	LLR-LOR	0.829	0.725	0.708	0.688
cosine	LLR	0.622	0.609	0.708	0.531
	LLR-MI	0.783	0.691	0.775	0.684

Table 2. Evaluation scores

each comparable corpus set described in Section 4.1 were examined. As listed in Table 2, combinations of the multiple measures did not achieve significant improvement on the Wikipedia corpus set. Meanwhile, the overall performance of Wikipedia did not significantly degrade in comparison with the newspaper set, although larger corpora give more appropriate associated word sets for each term pair. One reason for that result is the high comparability of Wikipedia data.

## 5 Concluding remarks

A novel method for constructing bilingual dictionaries via a third language is proposed. It applies a novel context similarity criterion, namely, a “weighted overlapping ratio” (WOR) for alleviating negative effects from translation perplexity. In addition, a method for combining word-association measures is developed. Experiments demonstrated the effectiveness of the proposed method: the proposed method achieved the highest  $F_{0.5}$ -score 0.729, thereby outperforming the  $F_{0.5}$ -score 0.691 by the conventional cosine-similarity method in the case of projecting context vectors onto English.

A future direction is applying word-sense-disambiguation techniques to associated words. By separating polysemous associated words into some classes corresponding to each sense, we could avoid the negative effect from unrelated senses of the associated words.

## Acknowledgments

This work was partially supported by Grant-in-Aid for Scientific Research, MEXT (22300032). We thank Prof. Masanori Kato for proofreading.

## References

- Bond, Francis, Ruhaida Binti Sulong, Takefumi Yamazaki, and Kentaro Ogura. 2001. Design and construction of a machine-tractable Japanese-Malay dictionary. In *Proceedings of MT Summit XIII*, pages 53–58.
- Daille, Béatrice and Emmanuel Morin. 2005. French-English terminology extraction from comparable corpora. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, Vol. 3651, pages 707–718.
- Déjean, Hervé and Éric Gaussier. 2002. Une nouvelle approche à l'extraction de lexiques bilingues à partir de corpus comparables. *Lexicometrica, Alignement lexical dans les corpus multilingues*, :1–22.
- Fišer, Darja, Špela Vintar, Nikola Ljubešić, and Senja Pollak. 2011. Building and using comparable corpora for domain-specific bilingual lexicon extraction. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 19–26.
- Fung, Pascale and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics*, Vol. 1, pages 414–420.
- Harris, Zellig. 1954. Distributional structure. *Word*, 10(23):146–162.
- Hazem, Amir, Emmanuel Morin, and Sebastian Peña Saldarriaga. 2011. Bilingual lexicon extraction from comparable corpora as metasearch. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora*, pages 35–43.
- Kaji, Hiroyuki. 2005. Extracting translation equivalents from bilingual comparable corpora. *IEICE Transactions on Information and Systems*, E88-D(2):313–323.
- Kaji, Hiroyuki and Toshiko Aizono. 1996. Extracting word correspondences from bilingual corpora based on word co-occurrence information. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 23–28.
- Kaji, Hiroyuki, Shin'ichi Tamamura, and Dashtseren Erdenebat. 2008. Automatic construction of a Japanese-Chinese dictionary via English. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 699–706.
- Laroche, Audrey and Philippe Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 617–625.
- Morin, Emmanuel and Emmanuel Prochasson. 2011. Bilingual lexicon extraction from comparable corpora enhanced with parallel corpora. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 27–34.
- Rapp, Reinhard. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, pages 320–322.
- Rapp, Reinhard. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 519–526.
- Sammer, Marcus and Stephen Soderland. 2007. Building a sense-distinguished multilingual lexicon from monolingual corpora and bilingual lexicons. In *Proceedings of MT Summit XI*, pages 399–406.
- Tanaka, Kumiko and Hideya Iwasaki. 1996. Extraction of lexical translations from non-aligned corpora. In *Proceedings of the 16th Conference on Computational Linguistics*, Vol. 2, pages 580–585.
- Tanaka, Kumiko and Kyoji Umemura. 1994. Construction of a Bilingual Dictionary Intermediated by a Third Language. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 297–303.
- Vulić, Ivan and Marie-Francine Moens. 2012. Detecting highly confident word translations from comparable corpora without any prior knowledge. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 449–459.
- Zhang, Yujie, Qing Ma, and Hitoshi Isahara. 2005. Construction of a Japanese-Chinese bilingual dictionary using English as an intermediary. *International Journal of Computer Processing of Oriental Languages*, 18(1):23–39.

# Two-Stage Pre-ordering for Japanese-to-English Statistical Machine Translation

Sho Hoshino<sup>1</sup> and Yusuke Miyao<sup>1,2</sup> Katsuhito Sudoh<sup>3</sup> and Masaaki Nagata<sup>3</sup>

<sup>1</sup>The Graduate University for Advanced Studies <sup>2</sup>National Institute of Informatics  
{hoshino, yusuke}@nii.ac.jp

<sup>3</sup>NTT Communication Science Laboratories, NTT Corporation  
{sudoh.katsuhito, nagata.masaaki}@lab.ntt.co.jp

## Abstract

We propose a new rule-based pre-ordering method for Japanese-to-English statistical machine translation that employs heuristic rules in two-stages. This two-stage framework contributes to experimental results that our method outperforms conventional rule-based methods in BLEU and RIBES.

## 1 Introduction

Reordering is an important strategy in statistical machine translation (SMT) to achieve high quality translation. While many reordering methods often fail in long distance reordering due to computational complexity, a promising technology called *pre-ordering* (Xia and McCord, 2004; Collins et al., 2005) has been successful for distant English-to-Japanese translation (Isozaki et al., 2010b). However, this strong effectiveness has not been shown for Japanese-to-English translation.

In this paper, we propose a novel rule-based pre-ordering method for the Japanese-to-English translation. The method utilizes simple heuristic rules in two-stages<sup>1</sup>: the inter-chunk and intra-chunk levels. Thus the method can achieve more accurate reorderings in Japanese. The translation experiments in patent domain showed that our method outperformed conventional rule-based methods, especially on the word reorderings. Our claims in this paper are summarized as follows:

1. The inter-chunk pre-ordering that relies on PAS analysis contributes to improvements in translation quality.
2. The intra-chunk pre-ordering which converts postpositional phrases into prepositional phrases further improves translation quality.
3. Thus, our two-stage framework is more effective than other pre-ordering methods.

<sup>1</sup>In this paper, we refer to Japanese *bunsetsu* as a “chunk”, a grammatical and phonological unit consists of noun, verb, or adverb followed by dependents such as particles.

## 2 Related Work

Japanese-to-English is challenging because the grammatical forms of the two languages are totally dissimilar. For instance, English is a head-initial language, and utilizes subject-verb-object (SVO) word orders, while Japanese is a *pure* head-final language, and utilizes subject-object-verb (SOV).

Komachi et al. (2006) proposed a rule-based pre-ordering method to convert SOV into SVO via a PAS analyzer. This method pre-orders inter-chunk level word orders in a single-stage, via the PAS analyzer which produced dependency trees and tagged each S, O, and V label. Then SOV sequences are converted into SVO. However, since the non-labeled words are left untouched, the effectiveness of this method is limited to simple SOV labeled matrix sentences without multiple clauses.

Katz-Brown and Collins (2008) proposed a two-stage rule-based pre-ordering method. In the first stage, SOV sequences are converted into SVO via the dependency analyzer. In the second stage, each chunk word order is naively reversed<sup>2</sup>.

Neubig et al. (2012) proposed a statistical model that was capable of learning how to pre-order word sequences from human annotated or automatically generated alignment data. However, this method has very large computational complexity to model long distance reordering.

## 3 Two-stage Pre-ordering Method

Here, we describe a new pre-ordering method which employs heuristic rules in two-stages. In the first stage, we reorganize and extend the rules described in (Komachi et al., 2006; Katz-Brown and Collins, 2008). In the second stage, we propose a new rule to consider chunk internal word orders. More precisely, **we apply four rules**: three rules for the first stage (Rule 1-1, 1-2, 1-3) and one

<sup>2</sup>Since the rule for S has not been described in detail, we provide a definition: S is a chunk followed by a topic-marker.

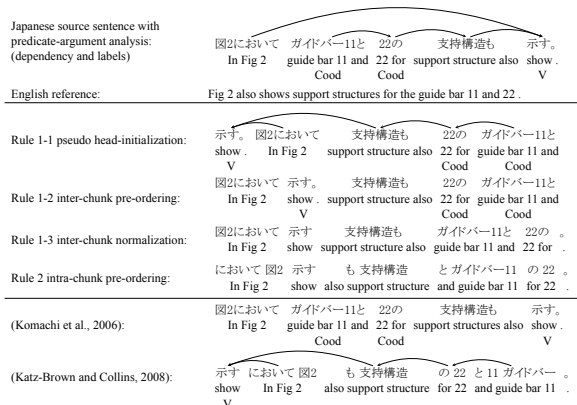


Figure 1: Pre-ordering Examples.

rule for the second stage (Rule 2). Each rule corresponds to the different linguistic nature between English and Japanese.

### 3.1 Rule 1-1 pseudo head-initialization

To output Japanese in head-initial sequences, this rule modifies Japanese dependency trees to the order that a head chunk comes first and its dependent children follow, by default. In the example, the sentence verbal head “示す show(s)” is moved to the leftmost position.

### 3.2 Rule 1-2 inter-chunk pre-ordering

This rule converts SOV into SVO. The rule can also handle a sentence which has no subject and object, due to a parsing error or a pro-drop that frequently occurs in Japanese. If there is V (a verbal head), then we apply this rule after Rule 1-1.

First, we move V instead of S or O, as we already have VSO sequences generated in Rule 1-1. Therefore, we placed V after the subject ( $V x^* S y^* \rightarrow x^* S V y^*$ ) or before the object in case a subject is not found ( $V x^* O y^* \rightarrow x^* V O y^*$ ).

Second, in the case where we only have V (without S and O), we move V before the rightmost chunk ( $V x^* y \rightarrow x^* V y$ ) to avoid head-initial outputs. In the example, since there is no subject and object, the verbal head “示す show(s)” is moved to immediately before its second dependent. On the other hand, V has been incorrectly placed in the rightmost in Komachi et al. (2006) and the leftmost in Katz-Brown and Collins (2008).

### 3.3 Rule 1-3 inter-chunk normalization

If there are coordinate clauses or punctuations, then we apply this rule after Rule 1-2. Basically,

we keep coordinate clauses and punctuations unchanged from the original word orders, by placing the coordinate clauses to the leftmost position and the punctuations to the rightmost position ( $x^* Punc y^* Cood z^* \rightarrow Cood x^* y^* z^* Punc$ ). In addition, in order to avoid comma-period sequences, we remove all commas immediately before a period.

In the example, the period “。” is moved to the rightmost position, unlike Komachi et al. (2006). And the coordinate clause “ガイドバー11と22の the guide bar 11 and 22” is restored to the original position in the source, by moving the clause to the rightmost position. While Katz-Brown and Collins (2008) does not have such a rule to restore the coordinate clause, Komachi et al. (2006) can keep it unchanged because that method does not move non-labeled words.

### 3.4 Rule 2 intra-chunk pre-ordering

For every chunk, we swap function and content words to organize pseudo prepositional phrases (Content Function  $\rightarrow$  Function Content). In the example, the chunk “ガイドバー11と the guide bar 11 AND” has three words: the two content words “ガイドバー11 the guide bar 11” and the function word “と AND”. Thus the chunk is reversed as “と ガイドバー11 AND the guide bar 11”.

### 3.5 Differences to Conventional Rules

Komachi et al. (2006) did not employ Rule 1-1 and only employed Rule 1-2. In this example, the head “support structures” should be followed by the dependents “guide bar 11 and 22”, but these words are left untouched. Katz-Brown and Collins (2008) employed Rule 1-1, the most of Rule 1-2, and a rule to keep punctuations as it partially treated by Rule 1-3. However, they did not have the exceptional rule to move verb from the first position for non-subject sentences, as described in Rule 1-2. In the example, this method misplaced the sentence verbal head “show” to the first position, and the coordination clause “guide bar 11 and 22” has also been mixed.

	BLEU	RIBES
Baseline	29.19	68.48
(Katz-Brown and Collins, 2008)	27.74	66.15
(Komachi et al., 2006)	29.58	69.10
(Neubig et al., 2012)	29.93	70.15
Proposed method	<b>30.65</b>	<b>72.26</b>

Table 1: Experimental Results.

Rule 1-2	Rule 1-3	Rule 2	BLEU	RIBES
			29.19	68.48
✓			29.76	71.00
	✓		27.71	69.50
		✓	28.29	65.61
	✓	✓	28.84	70.40
✓		✓	30.41	71.74
✓	✓		<b>30.94</b>	71.34
✓	✓	✓	30.65	<b>72.26</b>

Table 2: Ablation Tests.

	BLEU	RIBES
Proposed within KNP	<b>30.65</b>	72.26
Proposed within CaboCha+SynCha	30.01	<b>72.35</b>

Table 3: Differences in Parser Configurations.

## 4 Experiments

### 4.1 Experimental Setup

In order to compare pre-ordering methods, we conducted Japanese-to-English translation experiments on a fixed data set and SMT system.

For the common data set, we used the NTCIR-9 PatentMT Test Collection Japanese-to-English Machine Translation Data<sup>3</sup> package that contains approximately 3.2 million sentence pairs for training, 500 sentence pairs for development, and 2,000 sentence pairs for testing. The Japanese sentences are tokenized by MeCab 0.994<sup>45</sup>. In addition, we employed two parser configurations for Japanese parsing: (1) the KNP configuration used KNP 4.01<sup>6</sup> (Sasano and Kurohashi, 2011) for both dependency and PAS analyzer; (2) the CaboCha+SynCha configuration used CaboCha 0.65<sup>7</sup> (Kudo and Matsumoto, 2002) for dependency analysis and SynCha 0.3<sup>8</sup> (Iida and Poesio, 2011) for PAS analysis.

For the common SMT system, we used SRILM

<sup>3</sup><http://research.nii.ac.jp/ntcir/permission/ntcir-9/perm-en-PatentMT.html>

<sup>4</sup><http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>

<sup>5</sup>Since (unlike English) the Japanese language does not utilize spaces to delineate word boundaries, MeCab was used to perform the required Japanese tokenization.

<sup>6</sup><http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP>

<sup>7</sup><http://code.google.com/p/cabocho/>

<sup>8</sup><http://www.cl.cs.titech.ac.jp/~ryu-i/syncha/>

	BLEU	RIBES
Baseline	15.03	62.71
Proposed	<b>16.12</b>	<b>69.30</b>

Table 4: Results within a News Domain.

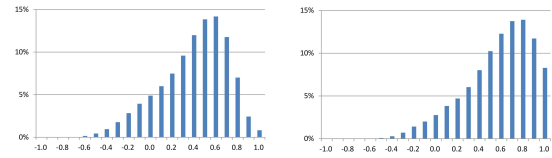


Figure 2: Kendall's  $\tau$  in Baseline and Proposed.

1.7.0 (Stolcke et al., 2011), MGIZA++ 0.7.3 (Gao and Vogel, 2008), Moses 0.91 (Koehn et al., 2007)<sup>9</sup>, and two popular evaluation metrics for Japanese-to-English: BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010a).

For the pre-ordering methods, we implemented rule-based methods proposed by (Komachi et al., 2006) and (Katz-Brown and Collins, 2008). In addition, an implementation<sup>10</sup> of statistical method proposed by Neubig et al. (2012) are used<sup>11</sup>. We did not use any pre-ordering in the baseline.

### 4.2 Experimental Results

Table 1 shows the experimental results for Japanese-to-English patent document translations that compare the following pre-ordering methods: the baseline (no pre-ordering), Komachi et al. (2006), Katz-Brown and Collins (2008), Neubig et al. (2012), our proposed method. We also conducted ablation tests, which consisted of a comparison of all Rule 1-2, 1-3, and 2, shown in Table 2.

The experimental results show that our proposed method outperformed all the other pre-ordering methods in terms of the BLEU and RIBES, which scored 30.65 and 72.26 points, respectively. This indicates that our two-stage pre-ordering method is better than conventional rule-based pre-ordering methods in the following aspects we found:

1. Our method and Komachi et al. (2006), both of which rely on PAS, were better than Katz-Brown and Collins (2008) which utilizes deterministic rules to obtain SOV labels.
2. Our intra-chunk pre-ordering gained a further improvement in translation accuracy as shown in Table 2. Nevertheless, we observed a 0.3 point drop in BLEU and a 0.9 point

<sup>9</sup>the following configurations are used in the system: 6-gram for language modeling, msd-bidirectional-fe for re-ordering, and MERT (Och, 2003) for tuning. After reviewing our preliminary findings, distortion limits were set to 20 for the baseline and (Komachi et al., 2006), and 10 for others.

<sup>10</sup><http://www.phontron.com/lader/>

<sup>11</sup>Only 10,000 sampled lines were used for training due to its computational complexity: During the training process, it consumed 120 GB of memory space for almost entire month.

improvement in RIBES by adding Rule 2 to Rule 1-2 and Rule 1-3, even though this combination yields better translations for native speakers. This phenomenon can be explained by the characteristic difference between BLEU and RIBES. While RIBES has a good correlation to human judgments, BLEU is said to have an uncorrelated, erratic behavior for Japanese-to-English translation (Isozaki et al., 2010a).

3. Our heuristic rules can cover more pre-ordering issues (as shown in Figure 1), and achieved further improvement in Rule 1-2 and Rule 1-3 as shown in Table 2.

In addition, as shown in Table 3, there was a 0.6 point statistically significant difference between two parser configurations (KNP and CaboCha+SynCha) in BLEU for our method. We suppose that one possible explanatory factor is the coordination structural accuracy to utilize Rule 1-3, because KNP tends to output more accurate coordination structures than CaboCha. However, it will be necessary to analyze our results in further detail to produce more definite conclusions. In any case, since such differences have been achieved simply by switching parsers, we believe that a better parsing method can be expected to produce better translation results in the future.

### 4.3 Pre-ordering Evaluation

We employed Kendall’s  $\tau$  rank correlation coefficient and its distribution as our pre-ordering criteria as described in Isozaki et al. (2010b). As shown in Figure 2, our proposed method produced much better correlation distribution than the baseline<sup>12</sup>.

Table 4 shows experimental results conducted on a news document that contains 150,000 sentence pairs created by (Utiyama and Isahara, 2003). Similar to the results shown in Table 1, our method outperformed the baseline.

## 5 Error Analysis and Discussion

Figure 3 shows an example within the proposed method. The intra-chunk rule Rule 2 moved the postposition “in” before the noun “Fig.7” and thus it makes the prepositional phrase “in Fig.7”. Also

<sup>12</sup>The average value of  $\tau$  in the proposed method is 0.575 against 0.391 in the baseline, and the percentages of sentences which have value of 0.8 or higher were 33.9% in our proposed method and 10.2% in the baseline. This 20% difference represents great reduction of word order differences.

Source: ここで、表1及び図7に示す各記号は、次のものを表している。 Here, Table 1 and Fig.7 in show each symbol, following things represent.
Reference: Here, symbols shown in Table 1 and Fig.7 represent the following items.
Proposed: でここは各記号示す及び表1に図7、いる表してをもの次の。 Here each symbol show and Table 1 in Fig.7, represent the things following.
Translated Result: In this case, the respective symbols shown in Table 1 and Fig.7 represents the followings.

Figure 3: Successful Pre-ordering Example

the coordination structure “Table 1 and Fig.7” is kept. There is still a minor verb agreement error which the verb “represent” is translated as “represents”. However, the most of errors are given via parsing process. For instance, of the first 30 sentences in the test data, we found 21 SOV tagging errors and 9 critical dependency errors, despite CaboCha is reported to have 89.8% accuracy for overall dependencies. Other methods could not translate this example correctly.

Besides, we also found that our deterministic rules cannot handle some difficult Japanese constructions. As a result, incorrect reordering had been conducted. For example, many Japanese sentences have a *topic* with a *topic-object-verb* construction, instead of subject-object-verb, because Japanese is a *topic-prominent* language. In the same 30 sentences, 18 sentences formed the topic-object-verb construction, and 4 sentences have been found as the topic-subject-object-verb construction.

## 6 Conclusion

In this paper, we proposed a new rule-based pre-ordering method for Japanese-to-English statistical machine translation, and we showed that our two-stage pre-ordering scheme was capable of solving more complex pre-ordering problems than conventional methods. From the experimental results, we found that our proposed method outperformed existing rule-based pre-ordering methods in terms of standard evaluation metrics.

## Acknowledgments

We would like to thank Ryu Iida, Mamoru Komachi, Taku Kudo, Graham Neubig, Ryohei Sasano, Wu Xianchao, and anonymous reviewers.

## References

- Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540.
- Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, June.
- Ryu Iida and Massimo Poesio. 2011. A cross-lingual ILP solution to zero anaphora resolution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 804–813.
- Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, October.
- Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for SOV languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251.
- Jason Katz-Brown and Michael Collins. 2008. Syntactic reordering in preprocessing for Japanese→English translation: MIT system description for NTCIR-7 patent translation task. In *Proceedings of the NTCIR-7 Workshop Meeting*.
- 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.
- Mamoru Komachi, Yuji Matsumoto, and Masaaki Nagata. 2006. Phrase reordering for statistical machine translation based on predicate-argument structure. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 77–82.
- Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, pages 63–69.
- Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 843–853.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Ryohei Sasano and Sadao Kurohashi. 2011. A discriminative approach to japanese zero anaphora resolution with large-scale lexicalized case frames. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 758–766.
- Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*.
- Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79.
- Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 508–514.

# Grammatical Error Correction Using Feature Selection and Confidence Tuning

Yang Xiang<sup>1†</sup>, Yaoyun Zhang<sup>1</sup>, Xiaolong Wang<sup>1‡</sup>, Chongqiang Wei<sup>1</sup>,  
Wen Zheng<sup>1</sup>, Xiaoqiang Zhou<sup>1</sup>, Yuxiu Hu<sup>2</sup>, Yang Qin<sup>1</sup>

<sup>1</sup>Key Laboratory of Network Oriented Intelligent Computation,  
Harbin Institute of Technology Shenzhen Graduate School, China

<sup>2</sup>South University of Science and Technology of China

<sup>†</sup>windseed@gmail.com <sup>‡</sup>wangxl@insun.hit.edu.cn

## Abstract

This paper proposes a novel approach to resolve the English article error correction problem, which accounts for a large proportion in grammatical errors. Most previous machine learning based researches empirically collected features which may bring about noises and increase the computational complexity. Meanwhile, the predicted result is largely affected by the threshold setting of a classifier which can easily lead to low performance but hasn't been well developed yet. To address these problems, we employ genetic algorithm for feature selection and confidence tuning to reinforce the motivation of correction. Comparative experiments on the NUCLE corpus show that our approach could efficiently reduce feature dimensionality and enhance the final  $F_1$  value for the article error correction problem.

## 1 Introduction

Grammatical errors in English are common in written issues especially for learners of English as a second language (L2 learners). As a result, automatic grammatical error correction (GEC) sprung up and has attracted more and more research attention recently. Among various error types, article errors account for a large proportion (over 12% in NUCLE) and are very difficult to be corrected.

Articles in the English language include indefinite article *a* and *an*, definite article *the* and zero article *empty* which means no article is used in this position. Articles are determiners of noun phrases which are indispensable in English grammar. Article errors are common in written English including wrong use, missing, and un-

necessary use of articles. For example, in the following sentence “*Over these years, it had helped humans to improve the accessibility in the forms of cards to gain access to certain places.*” there are two *thes* in which the first one is required but the second one is unnecessary. It is difficult for L2 learners to judge whether an article is necessary or not, or which article is needed. These errors are highly correlated with the context features around noun phrases. Errors occur frequently in various written issues which motivates researchers to exploit automatic error correction.

There are two main approaches for English article error correction. One of them is the external language materials based approach. Although there are minor differences on strategies, the main idea of this approach is to use frequencies such as n-gram counts as a filter and keep those phrases that have relatively high frequencies. Typical researches are shown by (Yi et al., 2008) and (Bergsma et al., 2009). Similar methods also exist in HOO shared tasks<sup>1</sup> such as the web 1TB n-gram features used by (Dahlmeier and Ng, 2012a) and the large-scale n-gram model in (Heilman et al., 2012). The other is machine learning based approach in which syntactic and semantic context features are utilized to train classifiers. Han et al. (2006) take maximum entropy as their classifier and apply some simple parameter tuning methods. Felice and Pulman (2008) present their classifier-based models together with a few representative features. Seo et al. (2012) invite a meta-learning approach and show its effectiveness. Dahlmeier and Ng (2011) introduce an alternating structure optimization based approach.

---

<sup>1</sup> <http://clt.mq.edu.au/research/projects/hoo/hoo2012>



As far as we know, most machine learning based approaches collect their features empirically and mainly depend on the feature selection of the classifiers which may bring about noises and increase the computational complexity when the feature dimensionality goes excessive. Moreover, discussions about the setting of threshold in classifiers are insufficient. Some work made simple adjustments on predicted thresholds after training their classification models like (Han et al., 2006; Dahlmeier and Ng, 2012a). Tetreault and Chodorow (2008) proceed from the different confidence of predicted categories which is similar to the approach employed in our work. We consider it is crucial to measure the differences between predicted scores of each category especially for GEC task on those documents with relatively high quality because in many cases, to keep the original form are actually the best choice.

In this paper, we focus on the machine learning based approach on error annotated corpus and propose a novel strategy to solve article error correction problem. Primarily, we extract a large number of related syntactic and semantic features from the context. With the help of genetic algorithm, a best feature subset is selected out which could greatly reduce the feature dimensionality. For each testing instance, according to the predicted confidence scores generated by the classifier, our tuning approach measures the trade-off between scores in order to enhance the confidence to a certain category. We didn't include any external corpora as references in our work which is to be further exploited. Experiments on NUCLE corpus show that our approach could efficiently reduce feature dimensionality and take full advantage of predicted scores generated by the classifier. The evaluation result shows our approach outperforms the state-of-the-art work (Dahlmeier and Ng, 2011) by 2.2% in  $F_1$  on this corpus.

There are two main contributions in our work: one is that we add feature selection before training and testing which reduces feature dimensionality automatically. The other is that we make use of the differences of confidence scores between categories and discuss about various tuning approaches which may affect the final performance.

The remainder of this paper is arranged as follows. The next section introduces feature extraction and selection. Section 3 describes model training and confidence tuning. Experiments and analysis are arranged in Section 4. Finally, we give our conclusion in Section 5.

## 2 Feature Extraction and Selection

We take article correction as a multi classification task. Three categories including *a/an*, *the* and *empty* are assigned to specify the correct article forms in corresponding positions (*a* and *an* are distinguished according to pronunciation of the following word). For training, developing and testing, all noun phrases (NPs) are chosen as candidates to be corrected. We extract related features based on the context of an NP and do feature selection afterwards.

### 2.1 Feature Extraction

A series of syntactic and semantic features are extracted with the help of NLP tools like Stanford parser (Klein and Manning, 2003), Stanfordner (Finkel et al., 2005) and WordNet (Fellbaum, 1999). We adopt syntactic features such as the surface word, word n-gram, part-of-speech (POS), POS n-gram, constituent parse tree, dependency parse tree, name entity type and headword; semantic features like noun category and noun hypernym. Some extended features are extracted based on them and some previous work (Dahlmeier and Ng, 2012b; Felice and Pulman, 2008).

Through feature extraction, we get over 90 groups of different features. After binarization, the dimensionality exceeds to about 350 thousand in which many features occur only once. We tried to prune all sparse features but found the performance fell off greatly while a manual deletion of several of them could instead improve the result. We infer that the sparse features may become useful when serving as an element of some feature subset which motivates us to carry out feature selection.

### 2.2 Feature Selection

Feature subset selection is conducted in this module to select out wrapped features. Genetic algorithm (GA) has been proven to be useful in selecting wrapped features in previous work (ElAlami, 2009; Anbarasi et al, 2010) and is applied in our work.

The features are encoded into a binary sequence in which each character represents one dimension. We use the number "1" to denote that this dimension should be kept while the number "0" means that dimension should be dropped in classification. A binary sequence such as "0111000...100" is able to denote a combination of feature dimensions. GA functions on the feature sequences and finally decides

which feature subsets should be kept. Following the steps of traditional GA, our approach includes generation of initial individuals, crossovers, mutations and selection of descendants for each generation.

The fitness function is the evaluation metric  $F_1$  described in §4.1. After feature selection, we reduced our feature dimensionality from 350 thousand to about 170 thousand which greatly reduced complexity in training. As expected, there are still a great number of sparse features left.

### 3 Training and Tuning

#### 3.1 Training Using Maximum Entropy

All noun phrases (NPs) are chosen as candidate instances to be corrected. For NPs whose articles are erroneous with annotations, the correct ones are their target categories, and for those haven't been annotated (error-free), their target categories are the observed articles. These NPs contain two basic types: *with* and *without* wrong articles. Two examples are shown below:

*with: #/empty big apples* ~ Category *empty*

*without: the United States* ~ Category *the*

For each category in *a*, *the*, and *empty*, we use the whole *with* instances and randomly take samples of *without* ones, making up the training instances for each category. We consider all the *with* samples useful because each of them has an observed wrong article which indicates that the correct article is easily misused as the wrong one. Different ratios of *with* : *without* are experimented in our work to see how much the number of *without* samples, which is mentioned in previous work (Dahlmeier and Ng, 2011), affects the result in our model.

Maximum entropy (ME) is employed for classification which has been proven to have good performance for heterogeneous features in natural language processing tasks. We have also tried several other classifiers including SVM, decision tree, and Naïve Bayes but finally found ME performs better.

#### 3.2 Confidence Tuning

ME returns with confidence of each category for a given testing instance. However, for different instances, the distributions of predicted scores vary a lot. In some instances, the classifier may have a very high predicted score to a certain category which means the classifier is confident enough to perform this prediction while for some other instances, two or more categories may

share close scores, the case of which means the classifier hesitates when telling them apart.

Our confidence tuning strategy (Tuning) on the predicted results is based on a comparison between the observed category and the predicted category. It is similar to the “thresholding” approach described in (Tetreault and Chodorow, 2008). The main idea of this confidence tuning strategy is: the selection between *keep* and *drop* is based on the difference between confidence of the predicted category and the observed category. If this difference goes beyond a threshold  $t$ , the prediction is proposed while if it is under  $t$ , we won't do any corrections. The confidence threshold is generated through hill-climbing in development data aiming at maximizing  $F_1$  of the result.

## 4 Experiments

### 4.1 Data Set and Evaluation Metrics

The NUCLE corpus (Dahlmeier and Ng, 2011) introduced by National University of Singapore contains 1414 essays written by L2 students with relatively high proficiency of English in which grammatical errors have been well annotated by native tutors. It has a small proportion of annotated errors which is much lower than other corpora. Only about 1.8% of articles contain errors in this corpus. The corpus provides the original texts as well as annotations which we make use of to generate training and developing samples. We divide the whole corpus into 80%, 10% and 10% for training, developing and testing to make our approach comparable with the previous work.

The performance is measured with precision, recall and  $F_1$ -measure where precision is the amount of predicted corrections that are also corrected by the manual annotators divided by the whole amount of predicted corrections. Recall has the same numerator with precision while its denominator is the amount of manually corrected errors.

### 4.2 Experiment and Analysis

In our experiments, we firstly compare the results of the baseline system (without GA and tuning, labeled as ME) and GA to see how much GA contributes to the performance. And also, we list the results of our initial strategy that all sparse features were deleted from the feature space (-SF). The comparisons are shown in Table 1 (all the *without* instances are used without sampling). The results show the effectiveness of GA and the usefulness of the sparse features.

Secondly, we tried several *with: without* ratios in the composition of training instances to see how much the selection of instances affects the result in our model. Figure 1 describes the comparative results under different ratios and the results with or without confidence tunings (discussed next).

Model	Prec.	Rec.	F <sub>1</sub>
ME(-SF)	4.29	66.67	8.05
ME	4.46	65.80	8.35
ME+GA	5.42	68.68	<b>10.06</b>
ME+Tuning(-SF)	13.33	26.17	17.66
ME+Tuning	15.85	28.20	20.03
ME+GA+Tuning	20.19	23.04	<b>21.53</b>

Table 1. Experiments on feature selection.

This experiment is conducted without the intervention of GA. Different from the conclusion in the previous work, we find that, there are not obvious differences between results under different ratios in our model. Before tuning, the differences are tiny, and after tuning, we believe it is mainly due to the advantage of the tuning strategy that eliminates the effects of randomness to a great extent. It is also interesting to see the improvement of F<sub>1</sub> always follows the increase of precision and decrease of recall which is good for the trend of correction without human intervention. We use all the *without* instances in our following experiments to avoid other randomness.

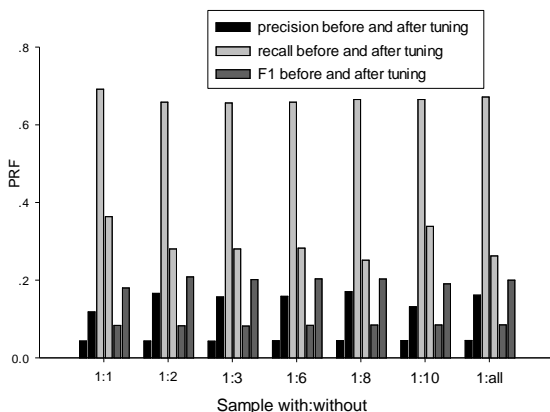


Figure 1. Comparisons before and after tuning. (*1:all* means to use the whole negative samples which is about *1:13*).

The best result of our model is achieved with GA and confidence tuning (ME+GA+Tuning in Table 1). Through experiments, we notice that the contribution of confidence tuning accounts for the largest proportion which directly enables our model outperform the previous state-of-the-

art work (precision of 26.44%, recall of 15.18%, and F<sub>1</sub> of 19.29% by (Dahlmeier and Ng, 2011)) by about 2.2% which is a big improvement in this task. Besides, the performance on the test set keeps that on the developing set which achieves precision of 20.97%, recall of 21.25%, and F<sub>1</sub> of 21.11%.

At last, we make comparisons on four threshold tuning strategies to verify the appropriateness of the thresholding approach applied in this paper. The five approaches labeled as *no-tuning*, *self-tuning*, *all-tuning*, *self-diff*, and *all-diff* in Table 2 correspond to the following four strategies. **(1)** Choose the category with the maximum predicted score; **(2)** Assign each category a fixed threshold beyond which a score goes most, that category is predicted; **(3)** Assign a fixed threshold for all categories beyond which a score goes most, that category is predicted; **(4)** Similar to (2) except that the threshold is the difference between scores of the predicted maximum and the observed category; **(5)** Similar to (3) except that the threshold is the difference between scores of the predicted maximum and the observed category.

Tuning method	Prec.	Rec.	F <sub>1</sub>
no-tuning	5.42	68.68	10.06
self-tuning	20.38	21.04	20.90
all-tuning	22.04	15.88	18.47
self-diff(our)	20.19	23.04	<b>21.53</b>
all-diff	22.82	17.00	19.49

Table 2. Different tuning strategies

It is noticeable that to assign a threshold for each category always performs better than to use a single threshold. We infer that the tuning strategies based on difference perform better mainly because they consider that the observed category should have a relatively high confidence if it is error-free even it is not the maximum.

## 5 Conclusion

In this paper, we introduce feature selection and confidence tuning for the article error correction problem. Comparative experiments show that our approach could efficiently reduce feature dimensionality and enhance the final F<sub>1</sub> value. However, for automatic grammatical error correction, there is still a long way to go. More resources and methods need to be exploited in the next stage for further performance improvement.

## Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 612-72383 and 61173075).

## References

- Anbarasi, M, E Anupriya, and NC Iyengar. Enhanced Prediction of Heart Disease with Feature Subset Selection Using Genetic Algorithm. *International Journal of Engineering Science and Technology*, Vol.2(10),2010: 5370-5376.
- Bergsma, S., D. Lin, and R. Goebel. 2009. Web-Scale Ngram Models for Lexical Disambiguation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.
- Dahlmeier, Daniel and Hwee Tou Ng. Grammatical Error Correction with Alternating Structure Optimization. In *Proceedings of the 49th Annual Meeting on Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2011.
- Dahlmeier, Daniel, Hwee Tou Ng, and Eric Jun Feng Ng. NUS at the HOO 2012 Shared Task. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2012a.
- Dahlmeier, Daniel and Hwee Tou Ng. A Beam-Search Decoder for Grammatical Error Correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*. Association for Computational Linguistics, 2012b.
- ElAlami, ME. A Filter Model for Feature Subset Selection Based on Genetic Algorithm. *Knowledge-Based Systems*, Vol.22(5), 2009: 356-362.
- Felice, Rachele De and Stephen G. Pulman. A Classifier-based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*. Association for Computational Linguistics, 2008.
- Fellbaum, C.. WordNet: An Electronic Lexical Data-base. *MIT Press*. 1998.
- Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2005.
- Han, N.R., M. Chodorow, and C. Leacock. Detecting Errors in English Article Usage by Non-native Speakers. *Natural Language Engineering*, Vol.12(02):115-129. 2006.
- Heilman, Michael, Aoife Cahill, and Joel Tetreault. Precision Isn't Everything: A Hybrid Approach to Grammatical Error Detection. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2012.
- Klein, Dan and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2003.
- Seo, Hongsuck et al. A Meta Learning Approach to Grammatical Error Correction. In *Proceedings of the 50th Annual Meeting on Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2012.
- Tetreault, Joel R. and Martin Chodorow. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*. Association for Computational Linguistics, 2008.
- Yi, X., J. Gao, and W.B. Dolan. 2008. A Web-Based English Proofing System for English as a Second Language Users. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*, 2008.

# An Online Algorithm for Learning over Constrained Latent Representations using Multiple Views \*

Ann Clifton and Max Whitney and Anoop Sarkar

School of Computing Science

Simon Fraser University

8888 University Drive

Burnaby BC, V5A 1S6, Canada

{ann\_clifton, mwhitney, anoop}@sfu.ca

## Abstract

We introduce an online framework for discriminative learning problems over hidden structures, where we learn both the latent structure and the classifier for a supervised learning task. Previous work on leveraging latent representations for discriminative learners has used batch algorithms that require multiple passes through the entire training data. Instead, we propose an online algorithm that efficiently jointly learns the latent structures and the classifier. We further extend this to include multiple views on the latent structures with different representations. Our proposed online algorithm with multiple views significantly outperforms batch learning for latent representations with a single view on a grammaticality prediction task.

## 1 Introduction

Natural language data is implicitly richly structured, and making use of that structure can be valuable in a wide variety of NLP tasks. However, finding these latent structures is a complex task of its own right. Early work used a two-phase pipeline process, in which the output of a structure prediction algorithm (e.g. a noun phrase finder) acts as fixed input features to train a classifier for a different task (e.g. grammaticality prediction). Chang et al. (2009), Das and Smith (2009), Goldwasser and Roth (2008), and Mccallum and Bellare (2005) have shown that this approach can propagate error from the structured prediction to the task-specific classifier. Recent work has combined unsupervised learning of (latent) structure prediction with a supervised learning approach for the task. Work in this vein has focused on jointly

\*This research was partially supported by an NSERC, Canada (RGPIN: 264905) grant and a Google Faculty Award to the third author.

learning the latent structures together with the task-specific classifier (Cherry and Quirk, 2008; Chang et al., 2010). Chang et al. (2010) in particular introduce a framework for solving classification problems using constraints over latent structures, referred to as Learning over Constrained Latent Representations (LCLR). We extend this framework for discriminative joint learning over latent structures to a novel online algorithm. Our algorithm learns the latent structures in an unsupervised manner, but it can be initialized with the model weights from a supervised learner for the latent task trained on some (other) annotated data. This can be seen as a form of domain adaptation from the supervised latent structure training data to the different classification task.

We evaluate our algorithm in comparison to the LCLR batch method on a grammaticality test using a discriminative model that learns shallow parse (chunk) structures. Our online method has standard convergence guarantees for a max-margin learner, but attains higher accuracy. Furthermore, in practice we find that it requires fewer passes over the data.

We also explore the use of allowing multiple views on the latent structures using different representations in the classifier. This is inspired by Shen and Sarkar (2005), who found that using a majority voting approach on multiple representations of the latent structures on a chunking task outperformed both a single representation as well as voting between multiple learning models. We show that the multiple-view approach to latent structure learning yields improvements over the single-view classifier.

## 2 The Grammaticality Task

To evaluate our algorithms, we use a discriminative language modeling task. A well-known limitation of  $n$ -gram LMs is that they are informed only by the previously seen word histories of a

fixed maximum length; they ignore dependencies between more distant parts of the sentence. Consider examples generated by a 3-gram LM:

- chemical waste and pollution control ( amendment ) bill , all are equal , and , above all else .
- kindergartens are now .

These fragments are composed of viable trigrams, but a human could easily judge them to be ungrammatical. However, if a language model used latent information like a shallow syntactic parse, it could also recognize the lack of grammaticality.

Discriminative models can take into account arbitrary features of data, and thus may be able to avoid the shortcomings of  $n$ -gram LMs in judging the grammaticality of text. In the case of language modeling, however, there is no obvious choice of categories between which the model should discriminate. Cherry and Quirk (2008) show that by following the pseudo-negative examples approach of Okanohara and Tsujii (2007), they can build a syntactic discriminative LM that learns to distinguish between samples from a corpus generated by human speakers (positives) and samples generated by an  $n$ -gram model (negatives).

Our approach is similar to Cherry and Quirk (2008), but they use probabilistic context-free grammar (PCFG) parses as latent structure, use a latent SVM as the learning model (we use latent passive-aggressive (PA) learning), and they handle negative examples differently. Instead of PCFG parsing, we use a chunking representation of sentence structure, which can be seen as a shallow parse, in which each word in the sentence is tagged to indicate phrase membership and boundaries.

Our model simultaneously learns to apply multiple sets of chunk tags to produce chunkings representing sentence structure and to prefer the shallow parse features of the human sentences to those sampled from an  $n$ -gram LM. The latent chunker will assign chunk structure to examples that yield the widest margin between the positive (grammatical) and negative (ungrammatical) examples.

### 3 Latent Structure Classifier

Our classifier is trained by simultaneously searching for the highest scoring latent structure while classifying data instances. Here we extend the latent learning framework due to Chang et al. (2010) from a batch setting to an online setting that uses passive-aggressive (PA) updates (Crammer and Singer, 2001).

### 3.1 PA Learning

The latent structure classifier training uses a decision function that searches for the best structure  $z_i^* \in Z(x_i)$  for each training sentence  $x_i$  with a space of possible structures  $Z(x_i)$  according to feature weights  $\mathbf{w}$ , i.e.:

$$f_w(x_i) = \arg \max_{z_i} \mathbf{w} \cdot \phi(x_i, z_i) \quad (1)$$

where  $\phi(x_i, z_i)$  is a feature vector over the sentence-parse pair. The sign of the prediction  $y_i^* \mathbf{w} \cdot \phi(x_i, z_i^*)$  determines the classification of the sentence  $x_i$ .

Using PA max-margin training (Crammer and Singer, 2001), we incorporate this decision function into our global objective, searching for the  $\mathbf{w}$  that minimizes

$$\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^X \ell(\mathbf{w} \cdot (f(x_i), y_i)), \quad (2)$$

where  $\ell$  is a loss function; we use hinge loss. At each iteration, for each example  $x_i$  we find and update according to a new weight vector  $\mathbf{w}'$  that minimizes:

$$\frac{1}{2} \|\mathbf{w} - \mathbf{w}'\|^2 + \tau(1 - y_i(\mathbf{w}' \cdot \phi(x_i, z_i^*))), \quad (3)$$

where  $\mathbf{w}$  is the previous weight vector,  $z_i^*$  is the structure found by Eqn. (1),  $y_i \in \{-1, 1\}$  is the example's true label (ungrammatical/grammatical), and  $\tau \geq 0$  is a Lagrange multiplier proportional to the example loss, thus penalizing classification examples in proportion to the extent that they violate the margin (see Alg. 1).

### 3.2 Optimization Method

Since Eqn. (3) contains an inner max over  $z_i^*$ , it is not convex for the positive examples, since it is the maximum of a convex function (zero) and a concave function  $(1 - y_i(\mathbf{w}' \cdot \phi(x_i, z_i^*)))$ . In hinge loss, driving the inner function to higher values minimizes the outer problem for negative examples, but maximizes it for the positives. So, as in LCLR, we hold the latent structures fixed for the positive examples but can perform inference to solve the inner minimization problem for the negatives.

### 3.3 Online Training

Our online training method is shown as algorithm 1. It applies the structured prediction and PA update of section 3 on a per-example basis in a variant of the cutting plane algorithm discussed in

```

1 initialize  $w_0$ 
2 for  $t = 0, \dots, T - 1$  do
3   for each training example  $x_i$  in  $X$  do
4     repeat
5       find  $z_i^* = \arg \max_{z_i} w_t \cdot \phi(x_i, z_i)$ 
6       let  $y_i^* = w_t \cdot \phi(x_i, z_i^*)$ 
7       let loss  $l_t = \max\{0, 1 - y_i y_i^*\}$ 
8       let multiplier  $\tau_t = \frac{l_t}{\|\phi(x_i, z_i^*)\|^2}$ 
9       update  $w_{t+1} := w_t + \tau_t y_i \phi(x_i, z_i^*)$ 
10    until  $y_i > 0$  or ( $y_i^* = y_i$  if  $y_i < 0$ );
11 return  $w_T$ 

```

Algorithm 1: Online PA algorithm for binary classification with latent structures.

Joachims and Yu (2009). Since for the positive examples the latent structures are fixed per-iteration, it does a single search and update step for each example at each iteration. For negative examples it repeats the prediction and PA update for each example until the model correctly predicts the label (i.e. until  $y_i^* = y_i$ ). Because of the intractability to compute all possible negative structures, we use the approximation of the single-best structure for each negative example. We re-decode the negative examples until the highest scoring structure is correctly labeled as negative. This approximation is analogous to the handling of inference over negative examples in the batch algorithm described in Chang et al. (2010). In the batch version, however, updates for all negative examples are performed at once and all are re-decoded until no new structures are found for any single negative example.

### 3.4 Multiple Views on Latent Representations

Shen and Sarkar (2005) find that using multiple chunking representations is advantageous for the chunking task. Moreover, they demonstrate that the careful selection of latent structure can yield more helpful features for a task-specific classifier. We thus perform inference separately to generate distinct latent structures for each of their five chunking representations (which are mostly from (Sang and Veenstra, 1999)) at line 5 of Alg. 1; at line 6 we evaluate the dot product of the weight vector with the features from the combined outputs of the different views.

Each of the views use a different representation of the chunk structures, which we will only briefly describe due to space limitations; for more detailed information, please see Shen and Sarkar (2005). Each representation uses a set of tags to label each token in a sentence as belonging to a non-overlapping chunk type. We refer to the chunking

Token	IOB1	IOB2	IOE1	IOE2	O+C
In	O	O	O	O	O
early	I	B	I	I	B
trading	I	I	I	E	E
in	O	O	O	O	O
Hong	I	B	I	I	B
Kong	I	I	E	E	E
Monday	B	B	I	E	S
,	O	O	O	O	O
gold	I	B	I	E	S
was	O	O	O	O	O
quoted	O	O	O	O	O
at	O	O	O	O	O
\$	I	B	I	I	B
366.50	I	I	E	E	E
an	B	B	I	I	B
ounce	I	I	I	E	E
.	O	O	O	O	O

Table 1: The five different chunking representations for the example sentence “In early trading in Hong Kong Monday , gold was quoted at \$ 366.50 an ounce .”

schemas as IOB1, IOB2, IOE1, IOE2, and O+C. The total set of tags for each of the representations are B- (current token begins a chunk), I- (current token is inside a chunk), E- (current token ends a chunk), S- (current token is in a chunk by itself), and O (current token is outside of any chunk). All chunks except O append the part-of-speech tag of the token as a suffix. Table 1 shows the different chunking schemas on an example sentence.

Each of these chunking schemas can be conceived as a different kind of expert. Of the inside/outside schemas, the IOB variants focus on detecting where a chunk begins; the IOE variants focus on the chunk’s end. O+C gives a more fine-grained representation of the chunking.

We use dynamic programming to find the best chunking for each representation. The features of  $\phi(x, z)$  are 1-, 2-, 3-grams of words and POS tags paired with the chunk tags, as well as bigrams of chunk tags. We use entirely separate chunk tags for each representation. E.g., although each representation uses an “O” tag to indicate a word outside of any phrase, we consider the “O” for each representation to be distinct.

We combine the multiple views in two different ways: 1) we simply concatenate the features from each structured prediction view into a larger feature vector and the weights are trained on the supervised learning task, and 2) before training on the supervised learning task we first convert all representations to a common representation, O+C (since it includes the union of the tagging distinctions from all 5 views, it does not cause loss of

information from any single view), and then we perform a majority vote for each tag in the prediction. We convert the winning sequence of predicted tags back to each representation and concatenate the features from each view as before and train on the supervised learning task.

## 4 Experiments

For the chunkers we used the CONLL 2000 tagset (23 chunk tags), modified for the five chunking representations of (Shen and Sarkar, 2005). We initialized the weights using a perceptron chunker. The chunker-classifier can either be started with a zero weight vector or with weights from training on the chunking task. For the latter, we used weights from supervised discriminative training against gold-standard chunking. To transfer the weights to the classifier, we scaled them to the range of values observed after training the zero-initialized chunker-classifier. For training data we used the English side of the HK Chinese-English parallel corpus, using 50,000 sentences as positive examples. For negative examples we used the pseudo-negative approach of Okanohara and Tsujii (2007): we trained a standard 3-gram language model on the 50,000 sentences plus 450,000 additional sentences from the same corpus. From this we sampled 50,000 sentences to create the negative training data set.

We evaluated the discriminative LMs on the classification task of distinguishing real grammatical sentences from generated pseudo-negative sentences. As test data we used the Xinhua data from the English Gigaword corpus. We used the first 3000 sentences as positive examples. For negative examples we trained a 3-gram LM on the first 500,000 examples (including those used for positive data). We used this 3-gram LM to generate five separate 3000 example negative data sets. To account for random variation due to using pseudo-negatives, results are reported as a mean over the positive data paired with each negative set. We evaluated our algorithms against LCLR as a baseline.<sup>1</sup> Table 2 shows that our online algorithm with

<sup>1</sup>We implemented two batch baselines. The first is a strict implementation of the LCLR algorithm as in Chang et al. (2010), with per-outer-iteration example caching (LCLR); we use a PA large-margin classifier instead of an SVM. However, we found that this algorithm severely overfits to our task. So, we also implemented a variant (“LCLR-variant”) that skips the inference step in the inner loop. This treated the latent structures from the inference step of the outer loop as fixed, but relabeled and updated accordingly until convergence, then resumed the next outer iteration.

Model	Accuracy %
LCLR	90.27
LCLR-variant	94.55
online single-view	98.75
+ multi-view	98.70
+ majority vote	98.78

Table 2: Classification accuracy after 40 outer iterations.

multiple views significantly outperforms the previous approaches. We omit a detailed experimental report of the behaviour of the online algorithm due to lack of space, but our findings were 1) that the batch models were slower to improve than the online versions on test-set accuracy, and 2) the online algorithm requires fewer updates total in training compared to the batch version.

## 5 Related and Future Work

As discussed, our work is most similar to Chang et al. (2010). We expand upon their framework by developing an efficient online algorithm and exploring learning over multiple views on latent representations. In terms of the task, max-margin LMs for speech recognition focus on the word prediction task (Gao et al., 2005; Roark et al., 2007; Singh-Miller and Collins, 2007). This focus is also shared by other syntactic LMs (Chelba and Jelinek, 1998; Xu et al., 2002; Schwartz et al., 2011; Charniak, 2001) which use syntax but rely on supervised data to train their parsers. Charniak et al. (2003) and Shen et al. (2010) use parsing based LMs for machine translation which are not whole-sentence models and they also rely on supervised parsers. Our focus is on using unsupervised latent variables (optionally initialized from supervised data) and training whole-sentence discriminative LMs. Our chunker model is related to the semi-Markov model in Okanohara and Tsujii (2007), but ours can take advantage of latent structures. Our work is related to Cherry and Quirk (2008) but differs in ways previously described.

In future work, we plan to apply our algorithms to a wider range of tasks, and we will present an analysis of the properties of online learning algorithms over latent structures. We will explore other ways of combining the latent structures from multiple views, and we will examine the use of joint inference across multiple latent representations.



## References

- Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Yuancheng Tu. 2009. Unsupervised constraint driven learning for transliteration discovery. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 299–307, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *HLT-NAACL*, pages 429–437.
- Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based Language Models for Machine Translation. In *Proc. of MT Summit IX*.
- Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proc. of ACL 2001*, pages 124–131, Toulouse, France, July. Association for Computational Linguistics.
- Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proc. of ACL 1998*, pages 225–231, Montreal, Quebec, Canada, August. Association for Computational Linguistics.
- Colin Cherry and Chris Quirk. 2008. Discriminative, syntactic language modeling through latent SVMs. In *Proc. of AMTA 2008*.
- Koby Crammer and Yoram Singer. 2001. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, January.
- Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. 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 - Volume 1*, ACL '09, pages 468–476, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jianfeng Gao, Hao Yu, Wei Yuan, and Peng Xu. 2005. Minimum sample risk methods for language modeling. In *Proc. of ACL*, pages 209–216, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Dan Goldwasser and Dan Roth. 2008. Transliteration as constrained optimization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 353–362, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Thorsten Joachims and Chun-Nam John Yu. 2009. Sparse kernel svms via cutting-plane training. *Machine Learning*, 76(2-3):179–193.
- Andrew McCallum and Kedar Bellare. 2005. A conditional random field for discriminatively-trained finite-state string edit distance. In *In Conference on Uncertainty in AI (UAI)*.
- Daisuke Okanohara and Jun'ichi Tsujii. 2007. A discriminative language model with pseudo-negative samples. In *Proc. of ACL 2007*, pages 73–80, Prague, Czech Republic, June. Association for Computational Linguistics.
- Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n-gram language modeling. *Computer Speech and Language*, 21(2):373–392.
- Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, EACL '99, pages 173–179, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lane Schwartz, Chris Callison-Burch, William Schuler, and Stephen Wu. 2011. Incremental syntactic language models for phrase-based translation. In *Proc. of ACL-HLT 2011*, pages 620–631, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Hong Shen and Anoop Sarkar. 2005. Voting between multiple data representations for text chunking. In *Canadian Conference on AI*, pages 389–400.
- Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-dependency statistical machine translation. *Comput. Linguist.*, 36(4):649–671.
- Natasha Singh-Miller and Michael Collins. 2007. Trigger-based Language Modeling using a Loss-sensitive Perceptron Algorithm. In *Proc. of ICASSP 2007*.
- Peng Xu, Ciprian Chelba, and Frederick Jelinek. 2002. A study on richer syntactic dependencies for structured language modeling. In *Proc. of ACL 2002*, pages 191–198, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

# Synonym Acquisition Using Bilingual Comparable Corpora

Daniel Andrade    Masaaki Tsuchida    Takashi Onishi    Kai Ishikawa

Knowledge Discovery Research Laboratories, NEC Corporation, Nara, Japan

{s-andrade@cj, m-tsuchida@cq,  
t-onishi@bq, k-ishikawa@dq}.jp.nec.com

## Abstract

Various successful methods for synonym acquisition are based on comparing context vectors acquired from a monolingual corpus. However, a domain-specific corpus might be limited in size and, as a consequence, a query term's context vector can be sparse. Furthermore, even terms in a domain-specific corpus are sometimes ambiguous, which makes it desirable to be able to find the synonyms related to only one word sense. We introduce a new method for enriching a query term's context vector by using the context vectors of a query term's translations which are extracted from a comparable corpus. Our experimental evaluation shows, that the proposed method can improve synonym acquisition. Furthermore, by selecting appropriate translations, the user is able to prime the query term to one sense.

## 1 Introduction

Acquiring synonyms or near synonyms is important for various applications in NLP, like, for example, paraphrasing and recognizing textual entailment (Bentivogli et al., 2009).

For these tasks, lexical resources like WordNet are often used to improve performance. Although these resources provide good coverage in general domains, they lack vocabulary specific to certain domains. Other problems are the limited availability and size of lexical resources for languages other than English.<sup>1</sup>

As a consequence, various previous works (Grefenstette, 1994) among others, suggest to acquire synonyms and other semantically

related words automatically from a monolingual corpus. The key assumption is that semantically similar words occur in similar context.

In general the larger the size of the monolingual corpus, the better and more detailed, we can extract the context, or context vectors for each relevant word (Curran and Moens, 2002). However, in a specific domain, the given monolingual corpus might be limited to a small size which leads to sparse context vectors. Another problem is that even for a specific domain, words can be ambiguous, which makes it unclear for which sense we are searching a synonym. For example, in the automobile domain, the ambiguous Japanese word バルブ (bulb, valve) has the synonyms 電球 (bulb) or 弁 (valve), depending on the meaning intended by the user.<sup>2</sup>

Our work tries to overcome both of these problems by enriching a context vector of a query word using *the context vectors of its translations* obtained from a comparable corpus in a different language. This way, some of the zero entries of a sparse context vector can be filled, and also disambiguation of the query word is possible. For example, if the desired query word is バルブ (bulb, valve), the user can select the auxiliary translation "valve" in order to mark the "valve" sense of the query word. Then, our system enforces the common parts of the context vector of バルブ (bulb, valve) and the context vector of "valve". Subsequently, when comparing the resulting context vector to synonym candidates' context vectors, the synonym 弁 (valve) will get a higher similarity score than the synonym 電球 (bulb).

In two experiments, we compare the proposed method to the baseline method which uses only the context vector obtained from the monolingual corpus. In the first experiment, the proposed method

<sup>1</sup>For example, the coverage of words for the English WordNet is 147,278 words, whereas for Japanese WordNet's coverage is only 93,834 words.

<sup>2</sup>To show the English meaning of a Japanese word, we place the English translations in brackets, directly after the Japanese word.

use all translations of a query term. In the second experiment, we use only the translations related to a certain sense of the query term. In both experiments the proposed method outperforms the baseline method, which suggests that our method is able to overcome sparsity and ambiguity problems.

In the following section we briefly embed our work into other related work. In Section 3, we explain our method in detail, followed by the two empirical evaluations in Section 4. We summarize our contributions in Section 5.

## 2 Related Work

Most work on synonym acquisition like (Grefenstette, 1994; Curran and Moens, 2002; Weeds and Weir, 2005; Kazama et al., 2010; Lin, 1998), contains basically of two steps: context vector extraction, and context vector comparison. In the first step, for the query term and each synonym candidate a context vector is extracted from the monolingual corpus. The context vector contains for example in each dimension how often the word co-occurred with another word in a certain syntactic dependency position. In the second step the query term's context vector is compared with each synonym candidate's context vector, for example by using the cosine similarity.

The problem of sparse context vectors, i.e. many dimensions in the context vector which contain zero entries, can be addressed by truncated Singular Value Decomposition and other matrix smoothing techniques (Turney and Pantel, 2010). We note that these smoothing techniques are complementary to our method since they could be applied after the context vector combination described in Section 3.2.

The additional use of bilingual (or multilingual) resources for synonym acquisition is also considered in (Van der Plas and Tiedemann, 2006) and (Wu and Zhou, 2003). Their work defines the context of word  $w$  in a certain sentence, as the translation of word  $w$ , in the corresponding translated sentence. However, their methods require bilingual (or multilingual) parallel corpora. For a word  $w$ , they create  $w$ 's context vector by using all word translations of  $w$ , wherein the word translations are determined by the word alignment in the parallel corpus. The weighting of each dimension of the context vector is determined by the number of times word  $w$  and its translation are aligned.

The methods described in (Hiroyuki and Morimoto, 2005; Li and Li, 2004) also use comparable corpora and word translations for disambiguating a certain query word. Their methods distinguish word senses by differences in word translations. For example, the senses of plant (factory, vegetation) are distinguished by the translations 工場 (factory) and 植物 (vegetation). Given a text snippet in which the ambiguous word occurs, their methods select the appropriate sense by finding an appropriate translation. In contrast, our method does not use a text snippet to disambiguate the meaning of the query word. Instead, our method uses one or more translations of the query word to find appropriate synonyms. For example, given the query word "plant" and the translation 工場 (factory) we expect to acquire synonyms like "manufacture", "factory" and so forth.

## 3 Proposed Method

We assume the user tries to find a synonym for the query term  $q$  in language A and provides additional translations of term  $q$  in language B. We name these translations as  $v_1, \dots, v_k$ . Furthermore we assume to have a pair of comparable corpora, one in language A and one in language B, and a bilingual dictionary.

We denote  $\mathbf{q}$  as the context vector of the term  $q$  in language A. A context vector  $\mathbf{q}$  contains in each dimension the degree of association between the term  $q$  and another word in language A which occur in the corpus written in language A. Therefore the length of context vector  $\mathbf{q}$  equals the number of distinct words in the corpus. We will use the notation  $\mathbf{q}(x)$  to mean the degree of association between the term  $q$  and the word  $x$  which is calculated based on the co-occurrence of term  $q$  and word  $x$  in the corpus.

We denote  $\mathbf{v}_1, \dots, \mathbf{v}_k$  as the context vectors of the terms  $v_1, \dots, v_k$  in language B. A context vector  $\mathbf{v}_i$ ,  $1 \leq i \leq k$ , contains in each dimension the degree of association between the term  $v_i$  and a word in language B.

### 3.1 Context Vector Translation

In the first step we estimate the translation probabilities for the words in language B to the words in language A for the words listed in the bilingual dictionary. For that purpose, we build a language model for each language using the comparable corpora, and then estimate the translation prob-

abilities using expectation maximization (EM) algorithm described in (Koehn and Knight, 2000). This way we get the probability that word  $y$  in language B has the translation  $x$  in language A, which we denote as  $p(x|y)$ .

We write these translation probabilities into a matrix  $T$  which contains in each column the translation probabilities for a word in language B into any word in language A. We use the translation matrix  $T$ , in order to translate each vector  $\mathbf{v}_i$  into a vector which contains the degree of association to words in language A. We denote this new vector as  $\mathbf{v}'_i$ , and calculate it as follows:

$$\mathbf{v}'_i = T \cdot \mathbf{v}_i \quad (1)$$

This way we get the translated context vectors  $\mathbf{v}'_1, \dots, \mathbf{v}'_k$ .

### 3.2 Context Vector Combination

In the second step, we combine the context vectors  $\mathbf{v}'_1, \dots, \mathbf{v}'_k$  and the context vector  $\mathbf{q}$ . Note that the dimension of a vector  $\mathbf{v}'_i$  and the vector  $\mathbf{q}$  is in general different, since  $\mathbf{v}'_i$  contains only the degree of association to the words listed in the bilingual dictionary.

We could now combine all context vectors additively, similar to monolingual disambiguation like in (Schütze, 1998). However, this would ignore that actually some dimensions are difficult to compare across the two languages. For example, it is difficult to translate the Japanese word *かける* (hang, put, bring,...) because of its many different meanings depending on the context. Therefore we combine the context vectors to a new context vector  $\mathbf{q}^*$  as follows: If a word  $x$  in language A is in the dictionary, we set

$$\mathbf{q}^*(x) := \mathbf{q}(x) + \sum_{i=1}^k \{(1 - c_x)\mathbf{q}(x) + c_x \cdot \mathbf{v}'_i(x)\}, \quad (2)$$

otherwise we set

$$\mathbf{q}^*(x) := (k + 1) \cdot \mathbf{q}(x). \quad (3)$$

$c_x \in [0, 1]$  is the degree of correspondence between word  $x$  and its translations in language B. The intuition of  $c_x$  is that, if there is a one-to-one correspondence between  $x$  and its translations, then we will set  $c_x$  to 1, and therefore consider the context vectors  $\mathbf{v}'_1$  and  $\mathbf{q}$  as equally important to describe the degree of association to word  $x$ . On

the other hand, if there is a many-to-many correspondence, then  $c_x$  will be smaller than 1, and we therefore rely more on the context vector of  $\mathbf{q}$  to describe the degree of association to word  $x$ . In case there is no translation available, we can rely only on the context vector of  $\mathbf{q}$ , and therefore set  $c_x$  to zero, see Formula (3).

Formally we set  $c_x$  as the probability that word  $x$  is translated into language B and then back into word  $x$ :

$$c_x = p(\bullet|x)^T \cdot p(x|\bullet) \quad (4)$$

where  $p(\bullet|x)$  and  $p(x|\bullet)$  are column vectors which contain in each dimension the translation probability from word  $x$  into the words of language B, and the translation probabilities from words in language B to word  $x$ , respectively. These translation probabilities are estimated like Section 3.1.

Finally, note that the vector  $\mathbf{q}^*$  is not rescaled. Depending on the vector comparison method, it might be necessary to normalize the vector  $\mathbf{q}^*$ . However, we will use in our experiments the cosine similarity to compare two context vectors, so the result does not change if we normalize or rescale  $\mathbf{q}^*$  by any non-zero factor.

## 4 Experiments

We extract synonyms from a corpus formed by a collection of complaints concerning automobiles compiled by the Japanese Ministry of Land, Infrastructure, Transport and Tourism (MLIT).<sup>3</sup> Our proposed method additionally consults a comparable corpus which is a collection of complaints concerning automobiles compiled by the USA National Highway Traffic Safety Administration (NHTSA).<sup>4</sup> The Japanese corpus contains 24090 sentences that were POS tagged using MeCab (Kudo et al., 2004). The English corpus contains 47613 sentences, that were POS tagged using Stepp Tagger (Tsuruoka et al., 2005), and use the Lemmatizer (Okazaki et al., 2008) to extract and stem content words (nouns, verbs, adjectives, adverbs).

For creating the context vectors, we use the co-occurrence counts of a word's predecessor and successor from the dependency-parse tree. These co-occurrence counts are then weighted using the

<sup>3</sup><http://www.mlit.go.jp/jidosha/carinf/rcl/defects.html>

<sup>4</sup><http://www-odi.nhtsa.dot.gov/downloads/index.cfm>

log-odds-ratio (Evert, 2004).<sup>5</sup> For comparing two context vectors we use the cosine similarity. The baseline method is the same as the proposed method except that it does not use Formula (2) and (3) to include information from the translations. As a bilingual dictionary, we use a large-sized Japanese-English dictionary with 1.6 million entries.<sup>6</sup>

In the first experiment we assume that the user wants to acquire all synonyms irrespectively of the difference in senses. Therefore, our gold-standard includes all words which occur in the corpus and which belong to any Japanese WordNet (Bond et al., 2009) synset to which the query term belongs. The gold-standard contains in total 234 Japanese words as query terms.<sup>7</sup> Our proposed method uses as auxiliary translations *all* English translations that correspond the query term and that are listed in our bilingual dictionary. For example, for the query term バルブ (bulb, valve), the proposed method uses the translations "bulb" and "valve".

The results in Table 1 (top) show that in average our method improves finding all synonyms for a query. The improvement can be accounted to the effect that our method enriches the sparse context vector of a Japanese query term.

In our second experiment, we assume that the user is interested only in the synonyms which correspond to a certain sense of the query term. For each query term we include into the gold-standard only the words belonging to one synset, which was randomly chosen. For example, for the ambiguous query term バルブ (bulb, valve) the gold-standard includes only the synset { 弁 (valve) }. That corresponds to a user looking for the synonyms of バルブ (bulb, valve) restricted to the sense of "valve". For selecting an appropriate translation, we use the cross-lingually alignment between the synsets of the Japanese and English WordNet (Bond et al., 2009; Fellbaum, 1998). Our proposed method will use as auxiliary translations only the query term's translations *that are listed in the corresponding English synset*. For example, for the query term バルブ (bulb, valve), the proposed method uses only the translation "valve".

The results in Table 1 (bottom) show a clear im-

<sup>5</sup>In preliminary experiments the log-odds-ratio best among other measures like point-wise mutual information, tf-idf and log-likelihood-ratio.

<sup>6</sup>This bilingual dictionary is not (yet) publicly available.

<sup>7</sup>Each query term as in average 2.3 synonyms which might correspond to different synsets in WordNet. In average, a query term's synonyms belong to 1.2 different synsets.

provement in recall by our proposed method. A pair-wise comparison of our proposed method and the baseline shows a statistically significant improvement over the baseline ( $p < 0.03$ ).<sup>8</sup> For example, we found that for the query term バルブ (bulb, valve), the baseline ranks 球 (bulb) at rank 3 and 弁 (valve) at rank 4, whereas the proposed method ranked 弁 (valve) at rank 3, and 球 (bulb) at rank 5. This suggests that our method can also help to disambiguate the context vector of an ambiguous query term.

All Senses					
Method	Top 1	Top 5	Top 10	Top 20	Inv. Rank
Baseline	0.10	0.24	0.29	0.37	0.29
Proposed	0.10	0.24	0.33	0.43	0.32
One Sense					
Method	Top 1	Top 5	Top 10	Top 20	Inv. Rank
Baseline	0.10	0.25	0.30	0.37	0.26
Proposed	0.10	0.26	0.35	0.45	0.30

Table 1: Recall at different ranks and inverse rank for gold-standard which considers all senses (top) and only one sense (bottom) for each query term. Recall at rank  $n$  is the number of correct synonyms which occur in the list from 1 to  $n$ , divided by all correct synonyms for a query. Inverse rank is the sum of the inverse ranks of each correct synonym for a query. All figures are the average over all query terms in the gold-standard.

## 5 Conclusions

We introduced a new method that combines a query term's context vector with the context vectors of the query term's translations acquired from a comparable corpus. This way our method is able to mitigate problems related to a query term's sparse context vector, and also helps to resolve its ambiguity.

The experiments showed that our method can improve synonym acquisition, when compared to a baseline method which does not use any comparable corpus.

We also demonstrated that our method can help to find the synonyms that are related to only one sense of the query term, by appropriately restricting the query term's translations. This way, our method can also be used to automatically populate resources like WordNet in languages different than English.

<sup>8</sup>We use the sign-test (Wilcoxon, 2009) to test the hypothesis that the proposed method ranks higher than the baseline.

## References

- L. Bentivogli, I. Dagan, H.T. Dang, D. Giampiccolo, and B. Magnini. 2009. The fifth pascal recognizing textual entailment challenge. *Proceedings of TAC*, 9:14–24.
- F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki. 2009. Enhancing the japanese wordnet. In *Proceedings of the 7th Workshop on Asian Language Resources*, pages 1–8. Association for Computational Linguistics.
- J.R. Curran and M. Moens. 2002. Scaling context space. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 231–238. Association for Computational Linguistics.
- S. Evert. 2004. The statistics of word cooccurrences: word pairs and collocations. *Doctoral dissertation, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart*.
- C. Fellbaum. 1998. Wordnet: an electronic lexical database. *Cambridge, MIT Press, Language, Speech, and Communication*.
- G. Grefenstette. 1994. *Explorations in automatic thesaurus discovery*. Springer.
- Kaji Hiroyuki and Yasutsugu Morimoto. 2005. Unsupervised word-sense disambiguation using bilingual comparable corpora. *IEICE transactions on information and systems*, 88(2):289–301.
- J. Kazama, S. De Saeger, K. Kuroda, M. Murata, and K. Torisawa. 2010. A bayesian method for robust estimation of distributional similarities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 247–256. Association for Computational Linguistics.
- P. Koehn and K. Knight. 2000. Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, pages 711–715. Association for the Advancement of Artificial Intelligence.
- T. Kudo, K. Yamamoto, and Y. Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 230–237. Association for Computational Linguistics.
- Hang Li and Cong Li. 2004. Word translation disambiguation using bilingual bootstrapping. *Computational Linguistics*, 30(1):1–22.
- D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics.
- N. Okazaki, Y. Tsuruoka, S. Ananiadou, and J. Tsujii. 2008. A discriminative candidate generator for string transformations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 447–456. Association for Computational Linguistics.
- H. Schütze. 1998. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.
- Y. Tsuruoka, Y. Tateishi, J. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. *Lecture Notes in Computer Science*, 3746:382–392.
- P.D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.
- L. Van der Plas and J. Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 866–873. Association for Computational Linguistics.
- J. Weeds and D. Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475.
- R.R. Wilcox. 2009. *Basic Statistics: Understanding Conventional Methods and Modern Insights*. Oxford University Press.
- H. Wu and M. Zhou. 2003. Optimizing synonym extraction using monolingual and bilingual resources. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 72–79. Association for Computational Linguistics.

# Exploring Verb Frames for Sentence Simplification in Hindi

Ankush Soni Sambhav Jain Dipti Misra Sharma

Language Technologies Research Centre

IIIT Hyderabad

{ankush.soni, sambhav.jain}@research.iiit.ac.in, dipti@iiit.ac.in

## Abstract

Systems processing on natural language text encounters fatal problems due to long and complex sentences. Their performance degrades as the complexity of the sentence increases. This paper addresses the task of simplifying complex sentences in Hindi into multiple simple sentences, using a rule based approach. Our approach utilizes two linguistic resources viz. verb demand frames and conjuncts' list. We performed automatic as well as human evaluation of our system.

## 1 Introduction

Cognitive and psychological studies, performed on 'human reading' states that the effort in reading and understanding a text increases with the sentence complexity (Klein and Kurkowski, 1974). The modern natural language processing applications are not much different, in this respect, from humans. Processing complex sentences with high accuracy has always been a challenge in computational linguistics. This calls for techniques aiming at automatic simplification of sentences (Chandrasekar et al., 1996).

The sentence complexity can be mainly classified into 'lexical complexity' and 'syntactic complexity'. In context of natural language applications, lexical complexity can be handled significantly by utilizing various resources like lexicons, dictionary, thesaurus etc. and substitute infrequent words with their frequent counterparts (De Belder et al., 2010). To address syntactic complexity, one can analyse the structure of the sentence and then apply proper operations to simplify the structure.

There are many applications of sentence simplification in NLP applications. Machine Translation systems when dealing with highly diverge language pairs face difficulty in translating long and complex sentences.

For Parsing, it has been shown by McDonald and Nivre(2007) that syntactic parsing of long sentence and Identifying long distance dependencies is still a challenging task for modern day parsers. So, it looks intuitive to break down the sentence into smaller parts and use the simplified sentences for the task of parsing and Machine translation.

In case of Automatic summarization, after simplifying sentences, it is likely that the accuracy of sentence extraction based summarization systems improves as smaller units of information are being extracted.

We present a rule based approach for sentence simplification in Hindi. Our proposed system takes a sentence and returns a set of simple sentences, smaller in length. We have taken care to produce sentences which keep the meaning close to the original sentence.

This paper is structured as follows: In Section 2, we discuss the related works for sentence simplification. Section 3 talks about complex sentence. Section 4 describes the linguistic resources we used. In section 5, we discuss our algorithm. Section 6 outlines evaluation of the system. In section 7, results are being talked about. Section 8 gives the error analysis and in Section 9, we conclude and talk about future works in this area.

## 2 Related Work

Chandrasekar et al.(1996) proposed Finite state grammar and Dependency based approach for sentence simplification. Automatic induction of rules for text simplification is discussed by Chandrashekar and Srinivas (1997). A pipelined approach for text simplification has been presented by (Siddharthan, 2002). Sudoh et al. (2010) proposed divide and translate technique to address the issue of long distance reordering for machine translation. Doi and Sumita (2003) used splitting techniques for simplifying sentences and then utilizing the output for machine translation. Poorn-

ima et al. (2011) proposed a rule based Sentence Simplification for English to Tamil Machine translation system.

Though several attempts, in the past, have been carried out for English, we find few work on other languages. We find, no reported work on sentence simplification for Hindi, which is the language under focus in our work.

### 3 Complex Sentence

Here we are addressing sentence complexity in the context to NLP applications, and our objective is to propose resolutions which could, in general, assist and improve the performance of the NLP systems. In general, complex sentences have more than one clause (Kachru, 2006) and these clauses are combined using connectives. In the context of dependency parsing, it has been illustrated by McDonald and Nivre(2007) that the sentence length increases the complexity of a sentence, as it is difficult to process on larger sentences. On experimenting for the Hindi language, we found that as the length of the sentence increases, number of verb chunks in the sentence also increases. Based on the above observation, we consider number of verb chunks as a criterion to define complex sentences. Also, we encounter the presence of conjuncts in long sentences and concede it as the second criterion representing a complex sentence.

To consolidate, for our approach we consider a sentence to be complex based on the following criteria:

- Criterion1 : Length of the sentence is greater than 5.
- Criterion2 : Number of verb chunks in the sentence is more than 1.
- Criterion3 : Number of conjuncts in the sentence is greater than 0.

Table 1 shows classification of a sentence based on the possible combinations of 3 criteria mentioned above.

### 4 Linguistic Resources

A list of conjuncts and verb frames form crucial resources for splitting a complex sentence into simple sentences.

Table 1: Classification of a sentence as simple or complex

Criterion1	Criterion2	Criterion3	Category
No	No	No	Simple
No	No	Yes	Simple
No	Yes	No	Simple
No	Yes	Yes	Simple
Yes	No	No	Simple
Yes	No	Yes	Complex
Yes	Yes	No	Complex
Yes	Yes	Yes	Complex

Table 2: verb-frame

arc-label	necessity	vibh(Case)	lex-cat	src-pos
<i>k1</i> (Doer)	mandatory	0	noun	1
<i>k2</i> (Experiencer)	mandatory	0	noun	1

#### 4.1 Connectives and Conjuncts List

Coordinating conjuncts are used to conjoin two independent clauses. Hindi coordinating conjuncts includes (*ora, athva, yaa, evam, para, magara, lekina, kintu, parantu, tatha, jabaki, va*). On the basis of the conjuncts joining two independent clauses we split the sentence for simplification.

#### 4.2 Verb Frames

Verb frames or verb subcategorization frames, categorizes the verb on the basis of their argument demands. For Hindi, verb frames have been discussed in Begum et al. (2008) . The verb frames show mandatory *karaka*<sup>1</sup> relation for a verb, i.e, the arguments of a verb. Verb demand frame is represented in a tabular form shown in Table 2. A verb frame shows :

1. *karaka* : dependency arc labels
2. *Necessity* of the argument ( mandatory(m) or optional(o) )
3. *Vibhakti* : post-position or the case associated with the nominal
4. *Lexical category* of the arguments.
5. *Position* of the demanded nominal with respect to verb (left(l) or right(r))

The Verb demand frames are built for the base form of a verb. The demands undergo a subsequent change based on the *tense, aspect* and *modality* (TAM) of the verb used in the sentence. The knowledge about the transformations induced on the base form of a verb by TAM is stored in

<sup>1</sup>*karakas* are the typed dependency labels in Computational Paninian Framework(Bharati and Sangal, 1993)



form of *transformation charts* for each distinct TAM.

## 5 Sentence Simplification Algorithm

We present a rule based method for simplification of complex sentences. Our approach comprises two stages. The work flow of our approach is shown in Figure 1. In the first stage, we get the structural representation of the input using shallow parser.

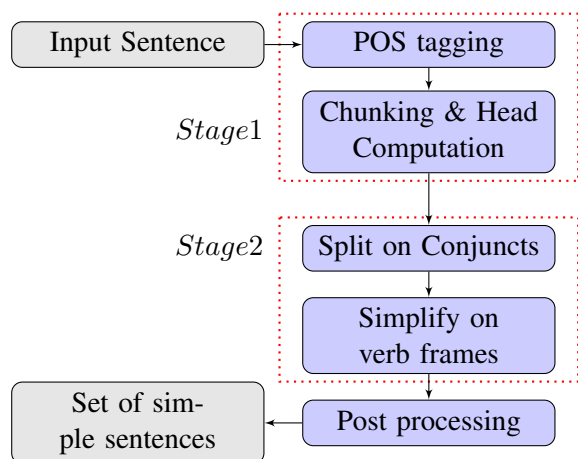


Figure 1: Flow-chart showing the work flow of sentence simplification system.

In the second stage, by applying predefined rules on the output of first stage, we identify the complexity in the sentence and simplify them on the basis of Conjuncts' list and Verb frames

### 5.1 Splitting on Conjuncts

In the first module, we split the sentence on the basis of Conjuncts. We identify the conjunct joining two independent clauses, break the sentence and pass it on to the second module for further simplification.

### 5.2 Simplification using Verb frames

After splitting the sentences on the basis of conjuncts, we simplify the generated sentences if they are complex. Once the type of sentence (complex or simple) is identified, multiple simple sentences are generated by converting non-finite verbs(VGNF) and gerunds(VGNN) to finite verb (VGF). Generally the arguments of VGNF and VGNN are shared with the main verb, therefore it is difficult for a machine to identify the implicit arguments of those verb and thus breaking the sen-

Table 3: karaka chart

arc-lbl	necessity	vibh	lex-cat	src-pos
k1	mandatory	0	noun	1
k2	mandatory	0	noun	1

tence and assigning arguments of those verbs explicitly helps in simplifying the sentence.

For conversion of VGNF to VGF, first, the head of the chunk (VGNF) is identified using shallow parser output. Verb frame of the root form of non-finite verb is used and transformations are carried out in accordance with the TAM of the finite verb of main clause. We follow the similar procedure in case of conversion from VGNN to VGF, with a difference that pronouns are generated in the place of VGNN.

Example of VGNF:

**Input:**

- (1) *ram khana khakar mohana ko bulata*  
 ram food eat-do mohana call  
*hai*  
 is

'After eating food, Ram calls for Mohana'

**Output:**

- (2) a. *ram khana khata hai*  
 ram food eat is  
 'Ram eats food'
- b. *ram mohana ko bulata hai*  
 ram mohana calls is  
 'Ram calls mohana'

In Input there is a VGNF (*khakar*), and needs to be converted to VGF

Here, in Input root word of *khakar* is '*kha*' and TAM of VGF in main clause is '*ta - hai*'. Verb frame of '*kha*' with '*ta*' as TAM is shown in Table 3:

Here, we can see that '*kha*' has 2 requirements 'k1' and 'k2' and both should be to the left of the verb as indicated by src-pos column. So, we will look for the argument to the left of the verb and accordingly form a sentence. For the verb '*kha*' in Input, '*Ram*' will act as k1 and '*khaana*' will act as k2. Now, we are left with a finite verb '*bulata*'. For the verb - '*bulata*' in Input, *Ram* will act as 'k1' and '*Mohana ko*' will act as 'k2'

with mandatory vibhakti 'ko'. As we can see here, *Ram* is the shared argument.

Example for VGNN

**Input:**

- (3) *karyasthalon para anusashana*  
workplaces at discipline  
*banae rakhna jaruri hai*  
maintain important is  
'It is important to maintain discipline at workplaces.'

**Output:**

- (4) a. *anusashana banana hai*  
discipline maintain is  
'Discipline is to be maintained'  
b. *karyasthalon para yah behad*  
workplaces at this very  
*jaruri hai*  
important is  
'This is very important at workplaces.'

Here '*banae rakhna*' is VGNN chunk with '*banana*' as verb and '*rakhna*' as auxiliary verb.

## 6 Evaluation

The evaluation of sentence simplification task is a difficult problem. The evaluation should address the following two factors: Readability (Adequacy and fluency) and Simplification. To consider these factors we perform both automatic as well as human evaluation.

### 6.1 Data

Our testing data set consists of 100 complex sentences taken randomly from the Hindi treebank (Bhatt et al., 2009; Palmer et al., 2009).

### 6.2 Automatic Evaluation

We used BLEU score (Papineni et al., 2002) for automatic evaluation of our system. Higher the BLEU score, closer the target set is to the reference set. The maximum attainable value is 1 while minimum possible value is 0.

For our Automatic evaluation we adopted the same technique as Specia (2010) using BLEU metric. We performed these 3 tests:

1. Computing BLEU Score between target set and reference set.
2. Computing BLEU Score between source set

Table 4: Bleu-score for the 3 data sets

System	Gold	Bleu-score
Target	Reference	0.805
Source	Reference	0.771
Target	Source	0.750

and reference set.

3. Computing BLEU Score between target set and source set.

## 6.3 Human Evaluation

To ensure the simplification quality subjective evaluation was done by human subjects. 20 sentences were randomly selected from the testing data-set of 100 sentences. Output of these 20 sentences, from the target set were manually evaluated by 3 subjects, who have done basic course in linguistics, for judging 'Readability' and 'Simplification' quality on the scale of 0–3, 0 being worst to 3 being the best for readability.

For Simplification performance, scores were given according to following criteria :

- 0 = None of the expected simplifications performed.
- 1 = Some of the expected simplifications performed.
- 2 = Most of the expected simplifications performed.
- 3 = Complete Simplification.

After taking input from all the participants the results are averaged out and shown in the section 7.2.

## 7 Results

### 7.1 Automatic Evaluation

Table 4 presents the result from automatic evaluation conducted on the lines of Specia (2010).

As it is evident from the results shown, that reference set matches more to target set (**0.805**) than to source set (**0.771**). From this we can conclude that simplification performed by our system is likely to be correct.

### 7.2 Human Evaluation

The readability and simplification score averaged over the three subjects is **1.85** and **2.07** respectively.

## 8 Error Analysis

Out of the 100 sentences put to test, 61 sentences are simplified by the system. 23 cases out of the unhandled cases were already simple as per our definition in section 3. On closer inspection we find 9 out of the remaining 16 unhandled cases are due to the presence of ‘complex predicates’. Complex predicates occur in form of nominal+verb combination and thus have a generative property. Due to their generative nature it is practically challenging to create verb demand frames for them. The remaining 7 cases are found to have POS and Chunking errors. On manually evaluating the output it was found that the quality of the output is affected by the dependency relations of arguments. The verb frame cannot capture the dependency of the required arguments thus leaving out few of the important dependencies.

## 9 Conclusion and Future Work

We present a rule based system for sentence simplification in Hindi. Our evaluation results show an average readability of **1.85** in the scale of 0-3, while **2.07** on the scale of 0-3 in system performance on simplification. Given the fact that this is the first attempt for Hindi we find our results satisfactory and have reason to believe that such a system will be beneficial in NLP Applications like parsing and MT. In the future our immediate effort would be on handling the complex predicates. We would like to try heuristics to capture the dependencies of the argument of verbs. We would also like to evaluate the impact of our tool on MT and parsing in the future.

## Acknowledgements

We would like to thank Kunal Sachdeva, Rahul Sharma, Rajesh Chaturvedi, Rishabh Srivastava and Riyaz Ahmad Bhat for their useful comments and feedback which helped to improve this paper.

## References

- Rafiya Begum, Samar Husain, D Sharma, and Lakshmi Bai. 2008. Developing verb frames in hindi.
- Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. Association for Computational Linguistics.
- R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D.M. Sharma, and F. Xia. 2009. A

multi-representational and multi-layered treebank for hindi/urdu.

- Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification.
- Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. Association for Computational Linguistics.
- Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. Lexical simplification.
- Takao Doi and Eiichiro Sumita. 2003. Input sentence splitting and translating.
- Yamuna Kachru. 2006. *Hindi*.
- Gary A Klein and Frank Kurkowski. 1974. Effect of task demands on relationship between eye movements and sentence complexity.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models.
- M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D.M. Sharma, and F. Xia. 2009. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure.
- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. Association for Computational Linguistics.
- C Poornima, V Dhanalakshmi, Anand M Kumar, and KP Soman. 2011. Rule based sentence simplification for english to tamil machine translation system.
- Advait Siddharthan. 2002. An architecture for a text simplification system. IEEE.
- Lucia Specia. 2010. Translating from complex to simplified sentences. Springer.
- Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Tsutomu Hirao, and Masaaki Nagata. 2010. Divide and translate: improving long distance reordering in statistical machine translation. Association for Computational Linguistics.

# Dirichlet Processes for Joint Learning of Morphology and PoS Tags

**Burcu Can**

Department of Computer Engineering  
Hacettepe University  
Beytepe, Ankara 06800 Turkey  
burcu.can@hacettepe.edu.tr

**Suresh Manandhar**

Department of Computer Science  
University of York  
Heslington, York, YO10 5GH, UK  
suresh.manandhar@york.ac.uk

## Abstract

This paper presents a joint model for learning morphology and part-of-speech (PoS) tags simultaneously. The proposed method adopts a finite mixture model that groups words having similar contextual features thereby assigning the same PoS tag to those words. While learning PoS tags, words are analysed morphologically by exploiting similar morphological features of the learned PoS tags. The results show that morphology and PoS tags can be learned jointly in a fully unsupervised setting.

## 1 Introduction

The morphology of a word is an important indicator that determines its PoS tag, meanwhile the PoS tag of a word helps in identifying the correct morphological segmentation of the word. This relationship between morphology and syntax has been beneficial in both morphology learning with the exploitation of the syntactic features and in PoS tagging with the adoption of morphological features.

There has been a number of research that have performed PoS tagging by making use of morphological information (Clark (2003), Hasan and Ng (2009), Abend et al. (2010), Christodoulopoulos et al. (2011), etc.). There has been also a number of other research that have performed morphological segmentation by adopting syntactic information (Hu et al. (2005), Can and Manandhar (2009), Lee et al. (2011), etc.). However, there is a small number of research that combines two tasks in a single framework.

Sirts and Alumäe (2012) share a similar goal

with us in joining PoS tagging and morphological segmentation in a single framework. They use hierarchical Dirichlet process for infinite HMMs to induce both PoS tags and morphological segmentation. Their model is type-based, whereas our model is token based. In our model, we use finite mixture models for PoS tagging and Dirichlet processes for segmentation.

## 2 Model Definition

The generative story of the model goes as follows:

1. Draw a PoS tag  $c_i$ .
2. Generate a word  $w_i$  that belongs to  $c_i$ .
3. Generate the context  $c_{i-1,i+1}$  of the word  $w_i$  from  $c_i$ .
4. From the possible splits of  $w_i$ , generate a suffix  $m_i$  conditioned on  $c_i$ , such that  $w_i = s_i + m_i$ , where  $s_i$  denotes the stem.

The generative story is summarised as follows:

$$p(c_i, c_{i-1,i+1}, w_i, s, m) = p(c_i)p(c_{i-1,i+1}|c_i) \\ p(w_i|c_i)p(m|c_i)p(s)$$

### 2.1 PoS Tagging

The model adopts a finite mixture model for PoS tagging (see Figure 1). Each mixture component represents a PoS tag that shares a set of features with other members in the same component. Each mixture component  $c_i$  consists of 1. a distribution over contexts and 2. a distribution over words. Each context is a PoS tag pair  $\langle c_{i-1}, c_{i+1} \rangle$  where the previous word  $w_{i-1}$  belongs to  $c_{i-1}$  and the following word  $w_{i+1}$  belongs to  $c_{i+1}$ . We employ a token-based approach for PoS tagging due to the significance of the context. The model is

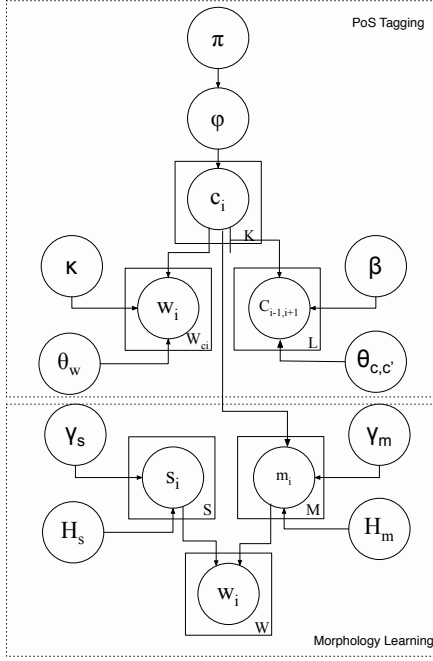


Figure 1: The complete joint model.

defined formally as follows:

$$c_i \sim \text{Mult}(\phi) \quad (1)$$

$$\phi \sim \text{Dir}(\pi) \quad (2)$$

$$w_i | c_i \sim \text{Mult}(\theta_w) \quad (3)$$

$$\theta_w \sim \text{Dir}(\kappa) \quad (4)$$

$$c_{i-1,i+1} | c_i \sim \text{Mult}(\theta_{c,c'}) \quad (5)$$

$$\theta_{c,c'} \sim \text{Dir}(\beta) \quad (6)$$

Class indicators  $c_i$  are drawn from a Multinomial distribution with parameters  $\phi$  (which have a Dirichlet prior distribution with hyperparameters  $\pi$ ). Each  $c_i$  involves a set of words  $w_i$  drawn from a Multinomial distribution with parameters  $\theta_w$  (which have a Dirichlet prior distribution with hyperparameters  $\kappa$ ). Each  $c_i$  also involves a distribution over contexts  $c_{i-1,i+1}$  drawn from a Multinomial distribution with parameters:  $\theta_{c,c'}$  (which have a prior distribution with hyperparameters  $\beta$ ).

## 2.2 Morphology Learning

We model morphology using a Dirichlet process (DP) in order to split each word into a stem and a suffix (see Figure 1). Stems are generated by  $DP(\gamma_s, H_s)$  with concentration parameter  $\gamma_s$  and base distribution  $H_s$ , whereas suffixes are generated by  $DP(\gamma_m, H_m)$  with concentration parameter  $\gamma_m$  and base distribution  $H_m$ . Hence, the

model is defined formally as follows:

$$s_i \sim DP(\gamma_s, H_s)$$

$$m_i | c_i \sim DP(\gamma_m, H_m)$$

Base distributions are length priors that favour shorter morphs (Creutz and Lagus, 2005):

$$H_x(x_i) = p(c_{ij})^{|x_i|} \quad (7)$$

where  $x_i$  is a morph and  $|x_i|$  is the length of  $x_i$  in letters. Each character has a probability of  $p(c_{ij})$ , where characters are assumed to be distributed uniformly in the alphabet. We also assume that each morph ends with a special character; i.e. end of morph marker.

Here,  $DP(\gamma_s, H_s)$  is a global Dirichlet process where stems may belong to any PoS tag, whereas  $DP(\gamma_m, H_m)$  is defined locally for each PoS tag. The reason is that stems are shared amongst different PoS tags. However, words belonging to the same PoS tag usually have similar endings, thereby leading to local distributions.

## 3 Inference

In our model, we assign values to the hyperparameters  $\pi, \kappa, \beta, \gamma_s, \gamma_m$  empirically, and we integrate out the parameters  $\phi, \theta_w, \theta_{c,c'}$  by using the Multinomial-Dirichlet conjugacy.

We use Gibbs sampling to infer POS tags, stems and suffixes. We perform inference in two steps: 1. a PoS tag is sampled for the word, 2. a stem and a suffix are sampled for the word.

### 3.1 Inferring PoS tags

Each word's PoS tag is sampled subject to its context. Let a word be  $w_i$  and imagine that it occurs in context  $\langle w_{i-1}, w_{i+1} \rangle$  where  $w_{i-1}$  belongs to  $c_{i-1}$  and  $w_{i+1}$  belongs to  $c_{i+1}$ . We define the sampling probability of  $c_i$  for  $w_i$  as follows:

$$p(c_i | \langle w_{i-1}, w_{i+1} \rangle, w_i) \propto \frac{p(\langle w_{i-1}, w_{i+1} \rangle, w_i | c_i) p(c_i)}{p(w_i | c_i) p(\langle w_{i-1}, w_{i+1} \rangle | c_i)} p(c_i)$$

We also assume that  $\langle w_{i-1}, w_{i+1} \rangle$  and  $w_i$  are independent since it is possible to remove  $w_i$  from  $\langle w_{i-1}, w_{i+1} \rangle$  and insert another word instead.

In order to calculate  $p(w_i | c_i)$ ,  $w_i$  is removed from the corpus:

$$p(w_i | c_i^{-w_i}, \kappa) = \frac{n_{w_i, c_i^{-w_i}} + \kappa}{N_{c_i}^{-w_i} + W_{c_i}^{-w_i} \alpha} \quad (8)$$

where  $c_i^{-w_i}$  denotes the mixture component  $c_i$  that excludes  $w_i$ ,  $n_{w_i, c_i^{-w_i}}$  is the number of the word-tag pairs  $\langle w_i, c_i \rangle$ ,  $N_{c_i}^{-w_i}$  is the number of word

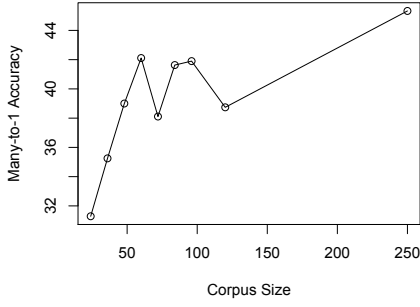


Figure 2: Many-to-1 accuracy scores obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K, and 250K.

tokens having the PoS tag  $c_i$ ,  $W_{c_i}^{-w_i}$  is the number of word types that are tagged with  $c_i$ .  $p(c_i)$  is computed as follows:

$$p(c_i | \mathbf{c}^{-w_i}, \pi) = \frac{n_{c_i}^{-w_i} + \pi}{N^{-w_i} + K\pi} \quad (9)$$

where  $N^{-w_i}$  denotes the number of word tokens in the model excluding  $w_i$ ,  $K$  is the number of class indicators (i.e. number of PoS tags).

In order to mitigate the sparsity within the context probabilities, we use the approximation introduced by Clark (2000):

$$p(\langle w_{i-1}, w_{i+1} \rangle | c_i) = \frac{p(\langle c_{i-1}, c_{i+1} \rangle | c_i)}{p(w_{i-1} | c_{i-1})p(w_{i+1} | c_{i+1})} \quad (10)$$

where,  $p(\langle c_{i-1}, c_{i+1} \rangle | c_i)$  is computed such that:

$$p(\langle c_{i-1}, c_{i+1} \rangle | c_x, c_y, c_z, c_i, \beta) = \frac{n_{c_{i-1}, c_i, c_{i+1}} + \beta}{k_{c_i} + L\beta} \quad (11)$$

Here,  $c_x$  is  $c_i^{-\langle c_{i-1}, c_{i+1} \rangle}$ ,  $c_y$  is  $c_{i-1}^{-\langle c_{i-2}, c_i \rangle}$ ,  $c_z$  is  $c_{i+1}^{-\langle c_i, c_{i+2} \rangle}$ ,  $k_{c_i}$  is the number of contexts in  $c_i$ , and  $L$  denotes the possible number of different contexts in the model (i.e.  $K * K$ ).

### 3.2 Inferring Morphology

Two latent variables are inferred for morphology: stems and suffixes. The sampling probability for morphology is defined as follows:

$$p(w_i = s_i + m_i | s^{-i}, \mathbf{m}_{c_i}^{-i}) = p(s_i | s^{-i})p(m_i | \mathbf{m}_{c_i}^{-i}) \quad (12)$$

where  $s^{-i}$  is the set of stems excluding  $s_i$ ,  $\mathbf{m}_{c_i}^{-i}$  is the set of suffixes assigned with  $c_i$  excluding  $m_i$ .

The conditional probability of a stem is:

$$p(s_i | s^{-i}, \gamma_s, H_s) = \frac{f^{s^{-i}} + \gamma_s H_s(s_i)}{T^{s^{-i}} + M^{s^{-i}} \gamma_s} \quad (13)$$

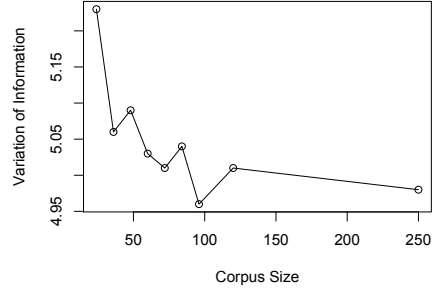


Figure 3: Variation of Information (VI) obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K and 250K.

where  $f^{s^{-i}}$  is the frequency of the stem type  $s_i$  already generated,  $T^{s^{-i}}$  is the number of all stems in the model, and  $M^{s^{-i}}$  is the number of stem types generated excluding  $s_i$ . Similarly, the conditional probability of a suffix is computed as follows:

$$p(m_i | \mathbf{m}_{c_i}^{-m_i}, \gamma_m) = \frac{f_{c_i}^{m^{-i}} + \gamma_m H_m(m_i)}{T_{c_i}^{m^{-i}} + M^{m^{-i}} \gamma_m} \quad (14)$$

where  $f_{c_i}^{m^{-i}}$  is the frequency of the suffix type  $m_i$  already generated in  $c_i$ ,  $T_{c_i}^{m^{-i}}$  is the number of all suffixes assigned with PoS tag  $c_i$ , and  $M^{m^{-i}}$  is the number of suffix types already generated excluding  $m_i$ .

In the algorithm, initially each word is assigned a PoS tag and split randomly. The algorithm goes through each word by sampling a PoS tag, a stem, and a suffix. All constituents of the respective word (tag, stem, suffix, context, contexts of adjacent words) are removed from the model beforehand. This process is repeated for a number of iterations until a convergence is ensured.

## 4 Experiments & Evaluation

We used small portions of the Penn WSJ treebank (Marcus et al., 1993) for the experiments. We manually set the hyperparameters and concentration parameters for each experiment:  $\pi = 10^{-6}$ ,  $\beta = 10^{-6}$ ,  $\kappa = 10^{-6}$ ,  $\gamma_s = 10^{-6}$ ,  $\gamma_m = 10^{-6}$ . These values were set empirically through several experiments. We also inserted a special character at the end of each sentence and assigned it a distinct PoS tag. No other words could be assigned this tag.

### 4.1 PoS Tagging Results

In our experiments we fixed the number of PoS tags to 45, which is the number of PoS tags in

	V-measure	Many-to-one
Christ.1 <sup>1</sup>	48.6	57.8
Joint	41.11	59.67
Clark <sup>2</sup>	63.8	68.8
Christ.2 (Best Pub.) <sup>3</sup>	67.7	72.0

<sup>1</sup> Christodoulopoulos et al. (2011)

<sup>2</sup> Clark (2003)

<sup>3</sup> Christodoulopoulos et al. (2010)

Table 1: PoS tagging scores.

	missing	extra	wrong	correct
Joint	0.72%	28.55%	10.13%	60.60%
Morfessor	15.07%	7.23%	10.22%	67.48%

Table 2: Morphological segmentation scores.

Penn WSJ treebank. We applied many-to-one accuracy by assigning each result tag a gold standard tag having the highest frequency among the words assigned with this result tag (see Figure 2). Second, we applied one-to-one accuracy which have similar results with many-to-one scores.

We also measured the variation of information (VI) (Rosenberg and Hirschberg, 2007) (see Figure 3). Although there is not a smooth decrement in VI measure, it improves with the larger datasets in average<sup>1</sup>.

Results show that determiners, modal verbs, prepositions, pronouns, conjunctions, and numbers are discovered generally correctly. The most common error type is due the confusion of nouns and adjectives. Normally, nouns are distributed over several PoS tags. Verbs and adverbs are also generally confused and spread over different tags.

We report our results with a comparison to other systems in Table 1 by using a dataset of 250K words. We use a small portion of Penn WSJ treebank for the comparison. The dataset involves 250K words where the number of word types is 20957. The other systems are also tested on a small portion of WSJ involving 16850 word types, which is reported in Christodoulopoulos et al. (2011).

Our system outperforms Christodoulopoulos et al. (2011) with the many-to-one evaluation, whereas Christodoulopoulos et al. (2011) perform better than our system based on V-measure evaluation. It should be noted that Clark (2003) and Christodoulopoulos et al. (2010) are both type-based.

<sup>1</sup>Although, Figure 3 shows that results for 36k words are better than results for 48k words, this could be due to the particular choice of training sets we used.

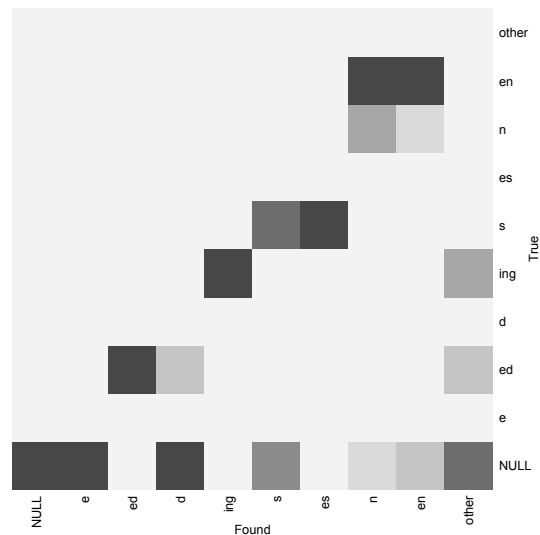


Figure 4: Confusion matrix shows the correlation between found morphs and true morphs. The shades reflect the number of matchings.

## 4.2 Morphological Segmentation Results

We performed the evaluation of morphological segmentation on verbs. We adopted some heuristics that strip off common verb endings such as *-ed*, *-d*, *-ing*, *-s*, *-es* from verbs in order to build the gold standard. Irregular verbs are introduced exceptionally and left as they are.

The results obtained from the 96K setting were used for the evaluation. We ran Morfessor Baseline (Creutz and Lagus, 2002; Creutz and Lagus, 2005; Creutz and Lagus, 2007) on the verbs in the same dataset. Table 2 gives the scores where *missing types* refers to the case that gold standard suggests a suffix but no suffix is identified in the results, *extra suffixes* means that gold standard does not identify any suffixes but the results contain suffixes, *wrong suffixes* implies that both gold standard and results identify suffixes but they are not the same, and *correct types* means that both gold standard and results contain suffixes and they match. Our model identifies 12257 suffix types, whereas Morfessor Baseline identifies 2309 due to undersegmentation. In addition, confusion matrix that depicts the result morphs against true morphs is given in Figure 4.

## 5 Conclusion

We proposed a model that jointly learns PoS tags and morphology. The results show that learning PoS tags and morphology can be performed cooperatively.

## References

- Omri Abend, Roi Reichart, and Ari Rappoport. 2010. Improved unsupervised pos induction through prototype discovery. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1298–1307, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Burcu Can and Suresh Manandhar. 2009. Clustering morphological paradigms using syntactic categories. In *Working Notes for the CLEF 2009 Workshop*, September.
- 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*, EMNLP '10, pages 575–584, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2011. A Bayesian mixture model for part-of-speech induction using multiple features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Alexander Simon Clark. 2000. Inducing syntactic categories by context distribution clustering. pages 91–94.
- Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 59–66, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL '02, pages 21–30, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mathias Creutz and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. *Technical Report A81*.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4:3:1–3:34, February.
- John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.
- Kazi Saidul Hasan and Vincent Ng. 2009. Weakly supervised part-of-speech tagging for morphologically-rich, resource-scarce languages. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 363–371, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yu Hu, Irina Matveeva, John Goldsmith, and Colin Sprague. 2005. Using morphology and syntax together in unsupervised learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, PMHLA '05, pages 20–27, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.
- Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Empirical Methods in Natural Language Processing*.
- Kairit Sirts and Tanel Alumäe. 2012. A hierarchical dirichlet process model for joint part-of-speech and morphology induction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 407–416, Stroudsburg, PA, USA. Association for Computational Linguistics.



# Parser Accuracy in Quality Estimation of Machine Translation: A Tree Kernel Approach

Rasoul Samad Zadeh Kaljahi<sup>†‡</sup>, Jennifer Foster<sup>†</sup>, Raphael Rubino<sup>†‡</sup>,  
Johann Roturier<sup>‡</sup> and Fred Hollowood<sup>‡</sup>

<sup>†</sup>NCLT, School of Computing, Dublin City University, Ireland  
{rkaljahi, jfoster, rrubino}@computing.dcu.ie

<sup>‡</sup>Symantec Research Labs, Dublin, Ireland  
{johann\_roturier, fhollowood}@symantec.com

## Abstract

We report on experiments designed to investigate the role of syntactic features in the task of quality estimation for machine translation, focusing on the effect of parser accuracy. Tree kernels are used to predict the segment-level BLEU score of English-French translations. In order to examine the effect of the accuracy of the parse tree on the accuracy of the quality estimation system, we experiment with various parsing systems which differ substantially with respect to their Parseval f-scores. We find that it makes very little difference which system we choose to use in the quality estimation task – this effect is particularly apparent for source-side English parse trees.

## 1 Introduction

Much research has been carried out on quality estimation (QE) for machine translation (MT) (Blatz et al., 2003; Ueffing et al., 2003; Specia et al., 2009; Callison-Burch et al., 2012), with the aim of solving the problem of how to accurately assess the quality of a translation without access to a reference translation. Approaches differ with respect to the nature of the quality scores being estimated (binary, 5-point or real-valued scales; human evaluations versus automatic metrics), the learning algorithms used or the feature set chosen to represent the translation pairs. The aspect of the task that we focus on is the feature set, and, in particular, the role of syntactic features. We ask the following: *To what extent is QE for MT influenced by the quality of the syntactic information provided to it? Does the accuracy of the parsing model used to provide the syntactic features influence the accuracy of the QE system?* We compare two pairs of parsing systems which differ with respect to their Parseval f-scores by around 17 absolute points in

a QE system for English-French MT and find that it makes little difference which system we use.

## 2 Related Work

Features extracted from parser output have been used before in QE for MT. Quirk (2004) uses a feature which indicates whether a full parse for a sentence can be found. Gamon et al. (2005) use part-of-speech (POS) tag trigrams, CFG production rules and features derived from a dependency analysis of the MT output. Specia and Giménez (2010) use POS tag language model probabilities of the MT output 3-grams. Hardmeier et al. (2012) combine syntactic tree kernels with surface features to produce a system which was ranked second in the WMT 2012 shared task on QE for MT (Callison-Burch et al., 2012). Rubino et al. (2012) explore source syntactic features extracted from the output of a hand-crafted broad-coverage grammar/parser and a statistical constituency parser. Avramidis (2012) builds models for estimating post-editing effort using syntactic features such as parse probabilities and label frequency. Like Hardmeier et al. (2012), we use tree kernels to represent the output of a parser, but unlike all the previous works, we explicitly examine the role of parser accuracy.

There have been some attempts to investigate the role of parser accuracy in downstream applications. Johannson and Nugues (2007) introduce an English constituency-to-dependency converter and find that syntactic dependency trees produced using this converter help semantic role labelling more than dependency trees produced using an older converter despite the fact that trees produced using the older converter have higher attachment scores than trees produced using the new converter. Mollá and Hutchinson (2003) find significant differences between two dependency parsers in a task-based evaluation involving an answer extraction system but bigger differences be-

tween the two parsers when evaluated intrinsically. Quirk and Corston-Oliver (2006) demonstrate that a syntax-enhanced MT system is sensitive to a decrease in parser accuracy obtained by training the parser on smaller training sets. Zhang et al. (2010) experiment with a different syntax-enhanced MT system and do not observe the same behaviour. Both Miyao et al. (2008) and Goto et al. (2011) evaluate a suite of state-of-the-art English statistical parsers on the tasks of protein-pair interaction identification and patent translation respectively, and find only small (albeit sometimes statistically significant) differences between the parsing systems. Our study is closest to that of Quirk and Corston-Oliver (2006) since we are taking one parser and using it to train various models with different training set sizes.

### 3 Parsing

For parsing we use the LORG parser (Attia et al., 2010)<sup>1</sup> which learns a latent-variable probabilistic context-free grammar (PCFG-LA) from a treebank in an iterative process of splitting the treebank non-terminals, estimating probabilities for the new rules using Expectation Maximization and merging the less useful splits (Petrov et al., 2006), and which parses using the max-rule parsing algorithm (Petrov and Klein, 2007).

In order to investigate the effect of parsing accuracy, we train two parsing models – one “higher-accuracy” model and one “lower-accuracy” model – for each language. We use training set size to control the accuracy. For English, the higher-accuracy model is trained on Sections 2-21 of the Wall Street Journal (WSJ) section of the Penn Treebank (PTB) (Marcus et al., 1994) (approx 40k sentences). For French, the higher-accuracy model is trained on the training section of the French Treebank (FTB) (Abeillé et al., 2003) (approx 10k sentences). For the lower-accuracy models, we first select four random subsets of varying sizes from the larger training sets for each language<sup>2</sup> and measure the performance of the resulting models on the standard parsing test sets<sup>3</sup> using Parseval  $F_1$  – see Table 1. All parsing models are trained with 5 split/merge cycles.

The worst-performing models for each language are those trained on 100 training sentences.

<sup>1</sup><https://github.com/CNGLdlab/LORG-Release>

<sup>2</sup>Each smaller subset is contained in all the larger subsets.

<sup>3</sup>WSJ Section 23 and the FTB test set.

However, these models fail to parse about 10 and 2 percent of our English and French data respectively. Since the failed sentences are not necessarily parallel in the source and translation sides, this could affect the downstream QE performance. Therefore, we opt to employ as our “lower-accuracy” models the second smallest training set sizes, which are 1K sentences for English and 500 for French. For both languages, the difference in  $F_1$  between the lower-accuracy and higher-accuracy models is about 17 points. In order to measure how different the parses produced by these models are on our QE data, we compute their  $F_1$  relative to each other. The  $F_1$  for the English model pair is 71.50 and for French 63.19.

### 4 Quality Estimation

To minimise the effect of domain variation, we use a QE dataset for the domain on which our parsers have been trained (newswire). Since there are very few human QE evaluations available for English-French in this domain, we instead attempt to predict automatic metric scores. We experiment with BLEU, METEOR and TER, but due to space restrictions and the similar behaviour observed, we report only BLEU score predictions. We randomly select 4500 parallel segments from the News development data sets released for the WMT13 translation task.<sup>4</sup> To remain independent of any one MT system, we translate the dataset with the following three systems, randomly choosing 1500 distinct segments from each:

- ACCEPT<sup>5</sup>: a phrase-based Moses system trained on training sets of WMT12 releases of Europarl and News Commentary plus data from Translators Without Borders (TWB)
- SYSTRAN: a proprietary rule-based system
- Bing<sup>6</sup>: an online translation system

The translations are scored at the segment level using segment-level BLEU. The data set is randomly split into 3000 training, 500 development, and 1000 test segments. Model parameters are tuned using the development set.

We encode syntactic information using tree kernels (Collins and Duffy, 2002; Moschitti, 2006) because they allow us to use all subtrees of the

<sup>4</sup><http://www.statmt.org/wmt13>

<sup>5</sup>[http://www.accept.unige.ch/Products/D\\_4\\_1\\_Baseline\\_MT\\_systems.pdf](http://www.accept.unige.ch/Products/D_4_1_Baseline_MT_systems.pdf)

<sup>6</sup><http://www.bing.com/translator>

Training size	English					French				
	100	<b>1K</b>	10K	20K	<b>40K</b>	100	<b>500</b>	2.5K	5K	<b>10K</b>
$F_1$	51.06	72.53	87.69	88.47	89.55	52.85	66.51	78.55	81.85	83.40

Table 1: Parser  $F_1$ s for various training set sizes: the sizes in bold are selected for the experiments.

parsed sentences as features in an efficient way, thus obviating the need for manual feature engineering. We use SVMLight-TK<sup>7</sup> (Moschitti, 2006), a support vector machine (SVM) implementation of tree kernels. The trees we use are constituency trees obtained by the parsing models described in Section 3, and their conversion to dependency trees using the Stanford converter for English (de Marneffe and Manning, 2008) and Const2Dep (Candito et al., 2010) for French. The labels must be removed from the arcs in the dependency trees before they can be used in SVMLight-TK – the nodes in the resulting tree representation are word forms and dependency relations, omitting part-of-speech tags.<sup>8</sup> Based on preliminary experiments on our development set, we use subset tree kernels.

We build a baseline system with features provided for the WMT 2012 QE shared task (Callison-Burch et al., 2012): we use Europarl v7 and News Commentary v8 (Koehn, 2005) to extract n-gram frequency, language model and word alignment features. This is considered a strong baseline as the system that used just these features was ranked higher than many of the other systems.

## 5 Experiments and Results

We build a QE system using constituency and dependency parse tree kernels of the source and translation sides, exploring first the higher-accuracy parse trees. Table 2 shows the performance of this system ( $CD-ST_H$ ) compared to the system trained on the baseline features ( $B-WMT$ ). We also compare to another baseline ( $B-Mean$ ) which always predicts the mean of the segment-level BLEU scores of the training instances. We evaluate performance using Root Mean Square Error (RMSE) and Pearson correlation coefficient ( $r$ ). To test the statistical significance of the performance differences (at  $p < 0.05$ ), we use paired bootstrap resampling (Koehn, 2004).

$CD-ST_H$  achieves statistically significantly bet-

<sup>7</sup><http://disi.unitn.it/moschitti/Tree-Kernel.htm>

<sup>8</sup>A word is a child of its dependency relation to its head and this dependency relation is the child of the head word.

ter RMSE and Pearson  $r$  than both baselines, which shows the usefulness of tree kernels in QE. We combine  $CD-ST_H$  and  $B-WMT$ <sup>9</sup> – this system  $B+CD-ST_H$  performs statistically significantly better than both systems individually, suggesting that tree kernels can also be useful in synergy with non-syntactic features.

	RMSE	Pearson $r$
B-Mean	0.1626	0
B-WMT	0.1601	0.1766
$CD-ST_H$	0.1581	0.2437
$B+CD-ST_H$	0.1570	0.2696

Table 2: Baselines, higher-accuracy parse tree kernels and combinations

We now investigate the impact of the intrinsic quality of the parse trees on the QE system. We build a similar model to  $CD-ST_H$  but with the lower-accuracy model described in Section 3. This system is named  $CD-ST_L$  in Table 3.  $CD-ST_H$  is also presented in this table for ease of comparison. Surprisingly,  $CD-ST_L$  performs only slightly lower than  $CD-ST_H$  and the difference is not statistically significant.

To better understand the behaviour of these systems, we break them down into their components: source constituency trees, target constituency trees, source dependency trees and target dependency trees. We first split based on the parse type and then based on the translation side.

$C-ST_H$  and  $C-ST_L$  in Table 3 are the systems with the constituency trees of both source and translation sides with higher- and lower-accuracy parsing models respectively. Although the difference is not statistically significant, the system with lower-accuracy parse trees achieves better scores than the system with higher-accuracy trees.  $D-ST_H$  and  $D-ST_L$  are built with the dependency trees of the higher- and lower-accuracy parsing models respectively. Unlike the constituency systems, the system with higher-accuracy parses performs better. However, the difference is not statistically significant. These results suggest that the intrinsic accuracy of neither the constituency

<sup>9</sup>The combination is carried out using vector summation.

nor the dependency parses is crucial to the performance of the QE systems. We now further split these systems based on the translation sides.

$C-S_H$  and  $C-S_L$  use the higher- and lower-accuracy constituency trees of only the source side. Similar to when constituency trees of both sides were used ( $C-ST_H$  and  $C-ST_L$ ), the system built on the lower-accuracy parses performs better although the difference is not statistically significant. The system using higher-accuracy constituency trees of the translation side ( $C-T_H$ ) achieves better scores than the one using the lower-accuracy ones ( $C-T_L$ ), but, again, this difference is not statistically significant.

$D-S_H$  and  $D-S_L$  are the systems using the dependency trees of only the source side. Again, there is a small, statistically insignificant gap between the scores of these systems. On the other hand, there is a bigger performance gap between the systems built on the higher- and lower-accuracy dependency trees of the translation side:  $D-T_H$  and  $D-T_L$ . Although this is the only large difference observed among all settings, it is surprisingly not statistically significant.<sup>10</sup>

	RMSE	Pearson r
CD-ST <sub>H</sub>	0.1581	0.2437
CD-ST <sub>L</sub>	0.1583	0.2350
C-ST <sub>H</sub>	0.1584	0.2307
C-ST <sub>L</sub>	0.1582	0.2348
D-ST <sub>H</sub>	0.1591	0.2103
D-ST <sub>L</sub>	0.1597	0.1902
C-S <sub>H</sub>	0.1583	0.2312
C-S <sub>L</sub>	0.1582	0.2335
C-T <sub>H</sub>	0.1608	0.1479
C-T <sub>L</sub>	0.1616	0.1204
D-S <sub>H</sub>	0.1598	0.1869
D-S <sub>L</sub>	0.1601	0.1780
D-T <sub>H</sub>	0.1598	0.2102
D-T <sub>L</sub>	0.1604	0.1679

Table 3: QE systems with higher- and lower-accuracy trees (C: constituency, D: dependency, ST: Source and Translation,  $H$ : Higher-accuracy parsing model,  $L$ : Lower-accuracy parsing model)

One may argue that the way the parser accuracy is varied here could impact the results – a parser with similar  $F_1$  but different output may lead to a different conclusion. It is possible to test this by using the parsing model from a lower split/merge (SM) cycle. For example, the models from the first SM cycle with a 10K training set size

<sup>10</sup>The high scores of  $D-T_H$  seem to be happening by chance, because on the development set, on which the parameters are tuned, the scores are much lower.

for English and a 2.5K training set size for French score 73.04 and 70.22  $F_1$  points on their respective test sets. While these scores are close to those of the lower-accuracy models used above, their outputs are different: the parses with the two lower-accuracy English models achieve only 66.46  $F_1$  against each other and with the two French ones 66.51  $F_1$ . We use the parse trees of these alternative lower-accuracy parsing models to build a new QE system. The RMSE is 0.1585 and Pearson r is 0.2316. These scores are not statistically significantly different compared to  $CD-ST_H$ , strengthening our conclusion that intrinsic parse accuracy is not crucial for QE.

Another question is to what extent we require a linguistically realistic syntactic structure which retains some form of regularity no matter how accurate. To answer this question, we build random tree structures for source and translation segments. The random tree for a segment is generated by recursively splitting the sentence into random phrases and randomly assigning them a syntactic label.<sup>11</sup> We parse the source and translation segments using this method and build a QE system with the output trees. The RMSE and Pearson r are 0.1631 and -0.0588 respectively. This shows that tree kernels still require the regularity encoded in the lower- and higher-accuracy trees.

## 6 Conclusion

We explored the impact of parse quality in predicting automatic MT evaluation scores, comparing the use of constituency and dependency tree kernels built from the output of parsing systems with a large accuracy gap when measured using Parseval  $F_1$ . This large difference in  $F_1$  did not have a knock-on effect on the QE task. Our next step is to carry out the experiments in the opposite direction (French-English) so that we better understand why the translation side trees were not as useful as the source side trees. Using other intrinsic parser evaluation metrics might also prove useful.

## Acknowledgements

This research has been supported by the Irish Research Council Enterprise Partnership Scheme (EPSPG/2011/102 and EPSPD/2011/135) and the computing infrastructure of the Centre for Next Generation Localisation at Dublin City University.

<sup>11</sup>The English random model achieves an  $F_1$  of around 0.5 and the French model an  $F_1$  of 0.2.

## References

- Anne Abeillé, Lionel Clément, and François Toussenet. 2003. Building a Treebank for French. In Anne Abeille, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Springer.
- Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling Unknown Words in Statistical Latent-Variable Parsing Models for Arabic, English and French. In *Proceedings of SPMRL*.
- Eleftherios Avramidis. 2012. Quality Estimation for Machine Translation Output Using Linguistic Analysis and Decoding Features. In *Proceedings of WMT*.
- John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence Estimation for Machine Translation. In *JHU/CLSP Summer Workshop Final Report*.
- 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 WMT*.
- Marie Candito, Benoît Crabbé, and Pascal Denis. 2010. Statistical French Dependency Parsing: Treebank Conversion and First Results. In *Proceedings of LREC'2010*.
- Michael Collins and Nigel Duffy. 2002. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proceedings of ACL*.
- Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford Typed Dependencies Representation. In *Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation*.
- Mollá Diego and Ben Hutchinson. 2003. Intrinsic versus Extrinsic Evaluation of Parsing Systems. In *Proceedings of EACL*.
- Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-Level MT Evaluation without Reference Translations: Beyond Language Modeling. In *Proceedings of EAMT*.
- Isao Goto, Masao Utiyama, Takashi Onishi, and Eichiro Sumita. 2011. A Comparison Study of Parsers for Patent Machine Translation. In *Proceedings of MT Summit*.
- Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Tree Kernels for Machine Translation Quality Estimation. In *Proceedings of WMT*.
- Richard Johansson and Pierre Nugues. 2007. Extended Constituent-to-dependency Conversion for English. In *Proceedings of NODALIDA*.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP*.
- Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit*.
- Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of ARPA Speech and Natural Language Workshop*.
- Yusuke Miyao, Rune Saetre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented Evaluation of Syntactic Parsers and their Representations. In *Proceedings of ACL*.
- Alessandro Moschitti. 2006. Making Tree Kernels Practical for Natural Language Learning. In *Proceedings of EACL*.
- Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of HLT-NAACL*.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact and Interpretable Tree Annotation. In *Proceedings of COLING-ACL*.
- Chris Quirk and Simon Corston-Oliver. 2006. The Impact of Parse Quality on Syntactically-informed Statistical Machine Translation. In *Proceedings of EMNLP*.
- Chris Quirk. 2004. Training a Sentence-Level Machine Translation Confidence Measure. In *Proceedings of LREC*.
- Raphael Rubino, Jennifer Foster, Joachim Wagner, Johann Roturier, Rasoul Kaljahi, and Fred Hollowood. 2012. DCU-Symantec Submission for the WMT 2012 Quality Estimation Task. In *Proceedings of WMT*.
- Lucia Specia and Jesús Giménez. 2010. Combining Confidence Estimation and Reference-based Metrics for Segment Level MT Evaluation. In *Proceedings of AMTA*.
- Lucia Specia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the Sentence-level Quality of Machine Translation Systems. In *Proceedings of EAMT*.
- Nicola Ueffing, Klaus Macherey, and Hermann Ney. 2003. Confidence Measures for Statistical Machine Translation. In *Proceedings of MT Summit*.
- Hao Zhang, Huizhen Wang, Tong Xiao, and Jingbo Zhu. 2010. The Impact of Parsing Accuracy on Syntax-based SMT. In *Proceedings of the International Conference on NLP-KE*.

# Attribute Relation Extraction from Template-inconsistent Semi-structured Text by Leveraging Site-level Knowledge

Yang Liu, Fang Liu, Siwei Lai, Kang Liu, Guangyou Zhou, Jun Zhao

National Laboratory of Pattern Recognition

Institute of Automation, Chinese Academy of Sciences

{yang.liu, fliu, swlai, kliu, gyzhou, jzhao}@nlpr.ia.ac.cn

## Abstract

A variety of methods have been proposed for attribute-value extraction from semi-structured text with consistent templates (strict semi-text). However, when the templates in semi-structured text are inconsistent (weak semi-text), these methods will work poorly. To overcome the template-inconsistent problem, in this paper, we proposed a novel method to leverage site-level knowledge for attribute-value extraction. First, we use a graph-based random walk model to acquire site-level knowledge. Then we utilize such knowledge to identify weak semi-text in each page and extract attribute-value pairs. The experiments show that, comparing to the baseline method which does not utilize site-level knowledge, our method can improve the extraction performance significantly.

## 1 Introduction

Among types of relations, attributes (e.g. nationality, date of birth) have emerged as one of the most popular types (Alfonseca et al., 2010), as they capture properties of respective objects (or instances) (e.g. *Kobe Bryant*). Generally, an attribute relation consists of an object, an attribute and its associated value (e.g. *Kobe Bryant - date of birth - August 23, 1978*, where “*August 23, 1978*” is the value of “*date of birth*”). In this paper, we call such a relation an object-attribute-value (OAV) tuple. Many methods have been proposed to extract attributes from semi-structured text (Cafarella et al., 2008)(Venetis et al., 2011)(Crescenzi et al., 2001)(Arasu and Garcia-Molina, 2003) and unstructured text like webpages and Web search query logs (Reisinger and Paşca, 2009)(Paşca et al., 2010)(Pasca and Van Durme, 2007). Semi-structured text (strict semi-text) often has distinctive HTML tags and consistent templates like

HTML tables (eg: Wikipedia infoboxes). However, a lot of user-generated semi-structured text with weak structures exist, where their templates generating records are inconsistent and the HTML tags in these templates are less distinctive. In this paper, we focus on the issue of extracting attribute-value (AV) pairs from semi-structured text with inconsistent templates (weak semi-text).

In previous work, Yoshinaga and Torisawa (Yoshinaga and Torisawa, 2007) extracted AV pairs of given objects from semi-structured text. They induced templates via a set of attributes obtained beforehand and used the templates to extract AV pairs. There are two constraints of their method. First, it heavily depends on the initial set of attributes. However, the quality and coverage of the initial set of attributes is hard to control. Second, they hold the assumption that attributes in the same block of semi-structured text are generated with the same template which weak semi-text does not satisfy. Their method mainly concentrated on extraction from semi-structured text with consistent templates (strict semi-text). When facing the weak semi-text with inconsistent templates, it will fail to obtain satisfactory results.

To resolve the problem of inconsistent templates, we propose an unsupervised method by leveraging site-level knowledge to extract AV pairs from weak semi-text. We explore the intrinsic structure connection among pages of the same website to address the problem. We make a two-stage effort: The first stage is to acquire knowledge that reveals the intrinsic similar structures among similar pages of the same site (site-level knowledge); the second stage is to leverage site-level knowledge to assist the AV pair extraction in weak semi-text.

In the paper, we present a novel approach that leverages site-level knowledge to extract instances’ attributes and their values from weak semi-text. To the best of our knowledge, little

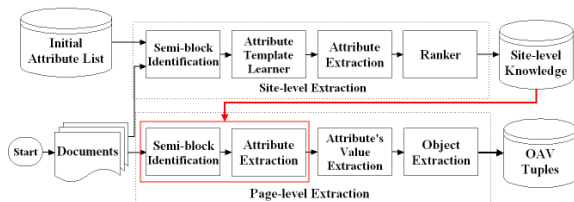


Figure 1: System overview.

work has addressed the problems to extract AV pairs from weak semi-text. The experimental results show that, when facing weak semi-text, our method outperforms the baseline method which does not leverage site-level knowledge.

## 2 System Description

The system consists of two parts: the site-level extraction and the page-level extraction (Figure 1). Site-level extraction aims to obtain site-level knowledge from pages of a website. Page-level extraction leverages obtained site-level knowledge to help the AV pair extraction from each page.

### 2.1 Site-level Extraction

We describe details of the modules in site-level extraction (Figure 1).

#### 2.1.1 Weak Semi-block Identification and Attribute Template Learner

We first segment a webpage into several blocks based on the paragraph HTML tags. Then we align the initial attributes to text of each block. The aligned attributes are used to induce templates to extract more attributes. A template is composed of a prefix and a separator. The separator is referred to the character or word next to the matched attribute and the prefix means characters previous to it. We take the string which begins at the head of first html tag before the matched attribute and ends at the head of the matched attribute as the template's prefix. For example a HTML fragment "...<div class="spctrl"></div> 性别(Sex): 男(Male)...", in it, "性别(Sex)" is the attribute, "</div> " is the prefix and ":" is the separator, the template is "</div> WC: " where WC is a placeholder for the attribute. And we set the prefix's window size as 15. If no html tag has been found within the window, then the template of this attribute is abandoned. Finally, we obtain a collection of templates of the weak semi-block.

We employ heuristic rules based on aligned attributes number and types and templates number

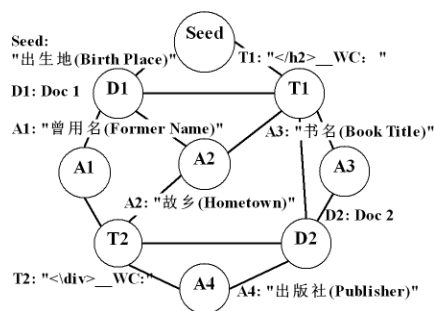


Figure 2: An example of our constructed graph.

to judge whether a block is a weak semi-block or not. Settings of two rules are discussed in the experiment at Section 5.4.1. They are: i) number of strings matched to initial attribute list is no smaller than  $t_1$ , and ii) sum of attributes' probabilities having been matched to strings in the text is larger than  $t_2$ . We represent them with a parameter vector  $T = (t_1, t_2)$ .

#### 2.1.2 Attribute Extraction

The obtained templates are used to extract more attributes in each block. Intuitively, more frequent a template is found in a weak semi-block, more likely a string extracted by that template is an attribute. Based on this idea, templates with higher frequencies will have higher priority than those with the lower frequencies when extracting attributes. After we run through all the pages of the site, we get a collection of templates and attributes. Then we rank them to obtain site-level knowledge.

#### 2.1.3 Ranker

To rank obtained templates and attributes to get site-level knowledge, we use the graph walk based technique (Wang and Cohen, 2007)(Wang and Cohen, 2009).

In the graph (Figure 2), attributes in initial attribute list are used as seeds. And these seeds are used to match the attributes in weak semi-block of a document (or a page) to learn templates. Then these templates are used to extract new attributes from the weak semi-block of a document (or a page). Intuitively, we consider that seeds appearing frequently are with high quality, templates derived by these seeds are tend to have good quality, and documents containing these seeds and templates are also deemed as high quality. Inversely, high quality documents also produce high quality attributes and high quality templates.

We utilize random walk with restart (RWR) to provide relevance score between two nodes (Tong et al., 2006). After the computation, we rank the attributes and templates by their probabilities in the final state vector.

We further refine the obtained ranked attributes by filtering obvious errors and the low ranks (site-level attributes) and generalize the top ranked templates by some rules (site-level templates). Site-level attributes and site-level templates composed the site-level knowledge.

## 2.2 Page-level Extraction

This section describes modules in page-level extraction (Figure 1).

### 2.2.1 Weak Semi-block Identification and AV Pair Extraction

To identify weak semi-block, we take the advantage of site-level knowledge to make several empirical rules based on the alignment of site-level templates and text of each block. The strings extracted by the templates are attribute candidates (*AttCandi* for short). We think only *AttCandies* extracted by authentic templates are correct attributes. A template is regarded as authentic once an *AttCandi* extracted by it exists in the site-level attributes. In the extraction of attribute’s values, we follow the method in (Yoshinaga and Torisawa, 2007) with the hypothesis that an attribute immediately precedes its value, and another AV pair immediately follows those values.

### 2.2.2 Object Extraction

we need to obtain **objects** of AV pairs to form attribute relations (**OAV tuples**) mentioned in Section 1 (eg: *Kobe Bryant - DateOfBirth - August 23, 1978*). We inspect several sampled pages and find their shared unique HTML template of objects for AV pair in their own pages. And then use this shared template to extract objects in each pages.

## 3 EXPERIMENTS

### 3.1 Experiment Settings

We carry out the experiments on 3 million Baidu Baike<sup>1</sup> (Baik for short) pages. In them, 1/3 of the pages (observed from our sampling) contain weak semi-text. For pre-processing, we remove infoboxes in each page which are strict semi-text.

<sup>1</sup><http://baike.baidu.com/>

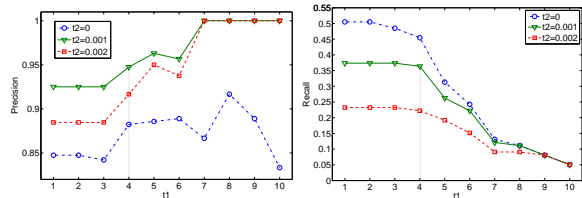


Figure 3: P/R curves with different  $T = (t1, t2)$ .

We evaluate on two aspects where site-level knowledge takes effect, they are: 1) weak semi-block identification in page-level extraction; 2) AV pair extraction in page-level extraction. We randomly sample 300 pages for manually labeling. 99 in them contain weak semi-blocks, and 1022 OAV tuples are labeled in the 99 pages. We use the manually labeled data as benchmark.

### 3.2 The Baseline

To demonstrate the effectiveness of incorporating the site-level knowledge, we implement a baseline system similar to Yoshinaga and Torisawa (Yoshinaga and Torisawa, 2007), which does not utilize the site-level knowledge. For comparison with our method, unlike their work which obtains initial attributes via search engine by manually generated regular expressions (it is hard to repeat precisely), we use the same initial attributes (attributes in infoboxes of Chinese Wikipedia) with our system as input.

### 3.3 Evaluation on Weak Semi-text

#### 3.3.1 Evaluation on Weak Semi-block Identification (Ours vs. Baseline)

For weak semi-block Identification, we vary parameter vectors  $T = (t1, t2)$  (Section 3.1) to show the selection of parameters. We set  $t1 = \{x : 1 \leq x \leq 10\}$ ,  $t2 = \{0, 0.001, 0.002\}$ . Details of their effects to precision curves and recall curves are shown in Figure 3.

Since the contradiction between precision and recall in figure 3, we think high precision is more important comparing to high recall. For that, if we fail to recall a weak semi-block, we still have chance to get the same features this weak semi-block contains from others in the same site and recall it when doing page-level extraction with the help of site-level knowledge, however, if we identify the incorrect weak semi-block, the incorrect knowledge in it will be added to site-level knowledge which will bring amount of errors to our results when utilizing it to help page-level extrac-



Table 1: Performances of weak semi-block location.

	Output number	Correct	Precision	Recall	F-measure
<i>Baseline</i> $T_\beta$	12	12	<b>1.0</b>	0.121	0.216
<i>Baseline</i> $T_\gamma$	59	50	0.847	0.505	0.633
<i>Baseline</i> $T_\alpha$	38	36	0.947	0.364	0.526
<i>SiteExt</i> $T_\alpha$	100	96	0.96	<b>0.970</b>	<b>0.965</b>

Table 2: Strict and loose precision (P), recall (R) and F-measure (F) comparison of OAV tuple acquisition.

	P-strict	R-strict	F-strict	P-loose	R-loose	F-loose
<i>Baseline</i> $T_\beta$	0.822	0.159	0.266	0.888	0.171	0.287
<i>Baseline</i> $T_\gamma$	0.691	0.356	0.470	0.736	0.380	0.501
<i>Baseline</i> $T_\alpha$	<b>0.856</b>	0.307	0.452	<b>0.918</b>	0.330	0.485
<i>SiteExt</i> $T_\alpha$	0.844	<b>0.770</b>	<b>0.805</b>	0.887	<b>0.810</b>	<b>0.847</b>

tion. Therefore, we choose  $T$  as  $T_\alpha = (4, 0.001)$ , for our system (**SiteExt**), which gives a relatively higher recall with a high precision (Figure 3).

We compared *SiteExt* $T_\alpha$  with  $T = T_\alpha$  and the baseline system which respectively uses  $T_\alpha$ ,  $T_\beta = (7, 0.001)$  and  $T_\gamma = (2, 0)$ . The weak semi-block identification module of the baseline system is the same with the weak semi-block identification module of SiteExt in site-level extraction (Section 3.2). Therefore the results in these two modules are the same. From Figure 3, we can see that *Baseline* $T_\beta$  brings the highest recall within the ones bringing highest precision, and *Baseline* $T_\gamma$  brings the highest precision within the ones bringing highest recall.

Table 1 shows that *SiteExt* $T_\alpha$ 's performance has a dramatic improvement comparing to other baseline systems which do not leverage site-level knowledge. The reason is that site-level knowledge captures attributes and templates specific to Baike. Meanwhile, weak semi-blocks in each page of the same site also share these features. As a result, we can identify more weak semi-blocks and reduce the incorrect ones with the same initial attribute set.

### 3.3.2 Evaluation on Object-Attribute-Value (OAV) tuples (SiteExt vs. Baseline)

We then evaluate the results of OAV tuple extraction. For different items in an OAV tuple, we select different similarity-computing methods. Because objects and attributes in an OAV tuple are always short phrases only with several words, we consider them as correct when their similarity meets a strict merit. On the other side, the value often contains descriptive contents which have more words. A small size of noises is acceptable. Therefore, besides the strict merit, we further select a loose

merit. The two merits are shown in (3) and (4).

$$S_{loose} = \frac{\text{len}(wd(V_{bm} \cap wd(V_{ext})))}{\min(\text{len}(wd(V_{bm})), \text{len}(wd(V_{ext})))} \quad (1)$$

$$S_{strict} = \frac{\text{len}(wd(V_{bm} \cap wd(V_{ext})))}{\max(\text{len}(wd(V_{bm})), \text{len}(wd(V_{ext})))} \quad (2)$$

Where  $V_{bm}$  and  $V_{ext}$  separately denote the string of an attribute's value in benchmark and in our extraction results,  $wd(V)$  is a set of different words in  $V$ , and  $\text{len}(s)$  means sum of words in a set  $s$ . In the experiment, we set the thresholds both as 0.75. When all the similarity scores of three items (object, attribute, value) exceed the threshold, the extracted OAV tuple is regarded as correct.

Table 2 shows the performance of different systems. Comparing to *Baseline* $T_\alpha$ , *SiteExt* $T_\alpha$  has great improvements in recall and has a slightly loss in precision. *SiteExt* $T_\alpha$  outperforms the other two baseline systems in both precision and recall. The experiment results prove that site-level knowledge is quite essential and effective to promise a good performance when extracting OAV tuples from weak semi-text of the same website. The two systems use the same initial attribute set as input, our method can identify more weak semi-blocks and extract more OAV tuples. It also proves that our method is less sensitive to the initial attribute set.

## 4 Conclusion

In this paper, we propose a novel approach that acquires site-level knowledge via a graph-based random walk model and leverages such knowledge to extract attribute relations from weak semi-text. Experimental results show that we can significantly improve the performance of identifying weak semi-text and OAV tuple extraction.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61070106, No. 61272332 and No. 61202329), the National High Technology Development 863 Program of China (No. 2012AA011102), the National Basic Research Program of China (No. 2012CB316300) and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201201).

## References

- E. Alfonseca, M. Pasca, and E. Robledo-Arnuncio. 2010. Acquisition of instance attributes via labeled and related instances. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 58–65. ACM.
- A. Arasu and H. Garcia-Molina. 2003. Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 337–348. ACM.
- M.J. Cafarella, A. Halevy, D.Z. Wang, E. Wu, and Y. Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.
- V. Crescenzi, G. Mecca, P. Merialdo, et al. 2001. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the international conference on very large data bases*, pages 109–118.
- M. Pasca and B. Van Durme. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837.
- M. Paşca, E. Alfonseca, E. Robledo-Arnuncio, R. Martin-Brualla, and K. Hall. 2010. The role of query sessions in extracting instance attributes from web search queries. *Advances in Information Retrieval*, pages 62–74.
- J. Reisinger and M. Paşca. 2009. Latent variable models of concept-attribute attachment. 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 620–628. Association for Computational Linguistics.
- H. Tong, C. Faloutsos, and J.Y. Pan. 2006. Fast random walk with restart and its applications.
- P. Venetis, A. Halevy, J. Madhavan, M. Paşca, W. Shen, F. Wu, G. Miao, and C. Wu. 2011. Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment*, 4(9):528–538.
- R.C. Wang and W.W. Cohen. 2007. Language-independent set expansion of named entities using the web. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 342–350. IEEE.
- R.C. Wang and W.W. Cohen. 2009. Character-level analysis of semi-structured documents for set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1503–1512. Association for Computational Linguistics.
- N. Yoshinaga and K. Torisawa. 2007. Open-domain attribute-value acquisition from semi-structured texts. In *Proceedings of the 6th International Semantic Web Conference (ISWC-07), Workshop on Text to Knowledge: The Lexicon/Ontology Interface (OntoLex-2007)*, pages 55–66.

# Optimum parameter selection for K.L.D. based Authorship Attribution for Gujarati

**Parth Mehta**

DA-IICT, Gandhinagar

parth.mehta126@gmail.com

**Prasenjit Majumder**

DA-IICT, Gandhinagar

prasenjit.majumder@gmail.com

## Abstract

We examine several quantitative techniques of authorship attribution that have gained importance over the time and compare them with the current state of the art Z-score based technique. In this paper we show how comparable the existing techniques can be to the Z-score based method, simply by tuning the parameters. We try to find the optimum values for number of terms, smoothing parameter value and the minimum number of texts required for creating an author profile.

## 1 Introduction

Authorship attribution and author profiling have a long standing history dating back to 19th century (Mosteller and Wallace, 1964). While authorship attribution deals with determining whether or not a given author has written the given article, author profiling aims at determining the age, gender, education level, etc. of the author from his article (Koppel et al., 2002). In the current work we focus only on authorship attribution. Authorship attribution techniques can be broadly classified into *linguistic techniques* and *Statistical techniques*. Until early 90's majority of the work done was from a linguistic perspective. Only after (Holmes, 1994) did statistical methods gain importance. There were many attempts to solve this particular problem using various statistics of texts like mean sentence length, term frequency distributions (Zipf, 1932), word lengths, character frequencies (Peng et al., 2003), vocabulary richness, etc. Rudman (1998) proposed nearly 1000 different measures that could be used. A whole new field of study called stylometry evolved from these type of studies. In 2002, (Burrows, 2002) proposed a novel technique based on Z-score that covered many of the above features, specially vocabulary difference and difference in term distribution

into a single measure. Later, Savoy(2010) modified this Z-Score that improved the result drastically and this technique is currently the state of the art for authorship attribution. In this paper we compare three major statistical techniques for authorship attribution to the state of the art Z-score based technique.

## 2 Corpus Details

The absence of a replicable and reliable corpora daunts the field of authorship attribution and more so for Indian languages. To the best of our knowledge the only work available to date for Indian languages is by (Bagavandas and Manimannan, 2008) where the corpus consisted of 55 Tamil articles, 32 of which were attributed and 23 disputed and there were only three possible authors. Having this fact in mind and following the steps of (Savoy, 2012) the authors developed a corpora consisting of 5039 newspaper articles from the newspaper *Gujarat Samachar*. These articles consist of 49 different weekly articles, from the supplements *Shatdal* and *Ravi Purti*, written by 40 distinct authors in the time period of 01-Dec-2011 to 1-Dec-2012 and is made available on our website<sup>1</sup>, along with the details of articles and authors. These articles span various categories like Science and Technology, short stories, Health and Fitness, Politics, etc. Though our corpus is more biased towards fiction(short stories & novels) this should not affect the performance because, unlike categories like health or art, stories seldom have a large overlap of vocabulary. Mean length of the documents was found to be 972 (Minimum: 85 Maximum: 1774, Median: 909,Standard deviation: 382). These texts were all available in the standard UTF-8 format and hence the only pre-processing that we did was to remove the punctuations, numerals and author names from the text. Except this no other pre-

<sup>1</sup><http://irlab.daiict.ac.in/tools.php>

processing was done, each morphological variant was treated as a unique term and also there was no word sense disambiguation to distinguish between same words having different meaning. Concept of capitalization does not exist as such in Gujarati language. Our experiments being completely statistical in nature can be replicated very easily without any knowledge of Gujarati language.

### 3 Experiment details

We compare four different Authorship attribution methods mentioned in (Savoy, 2012), namely Delta method, Chi-squared method, Z-Score based method and Kullback Leibler divergence based method. Our aim is to examine whether or not tuning the parameters of K.L.D. based method can produce results comparable to the state of the art Z-score based method. In this section we briefly describe each of the four methods and then explain the parameters that can affect K.L.D. based authorship attribution. All these methods are profile based methods i.e. for each of the  $N$  authors we create an author profile  $A_k$  where  $k \in \{1, \dots, N\}$ . These profiles are created from the documents for which the true author is already known. Disputed document  $Q$  is then compared to each author profile  $A_k$  using a metric  $D(Q, A_k)$  and is attributed to that author for whom  $D$  is minimum. In other words for given query text  $Q$  and author profiles  $A_k$

$$A_{correct} = \underset{k \in N}{\operatorname{argmin}} \{D(Q, A_k)\} \quad (1)$$

The distance function  $D$  depends on the method used and is defined separately for each method and so is true for the author profile  $A_k$ .

The parameters in these experiments that are to be set heuristically include the value of the smoothing technique and smoothing parameter ( $\lambda$ ) for that technique, the minimum number of texts( $N$ ) that have to be used in order to create a reasonably good author profile and the number of terms( $X$ ) considered to create the author and document profiles. Due to several studies readily available, we directly use Lidstone smoothing technique without further experimentation. Our main aim is to find the optimum value of these parameters for a corpus of Gujarati articles.

### 3.1 Delta Method

Delta method was first proposed by (Savoy, 2012). It uses a term-document index along with Z-score defined by equation 2 below

$$Z_{score}(t_{ij}) = \frac{tf_{ij} - mean_i}{sd_i} \quad (2)$$

Z-score is calculated for each term  $t_{ij}$  where  $i \in \{1, \dots, T\}$  and  $j \in \{1, \dots, M\}$ .  $T$  and  $M$  are the total number of unique terms and total number of documents in the corpus respectively.  $tf_{ij}$  is the term frequency of term  $i$  in document  $j$ ,  $mean_i$  and  $sd_i$  are the mean and standard deviation of frequency of term  $t_i$  in the entire corpus. Using this we can represent each document as a vector of Z-scores for each of its terms. Hence each document can be represented as a vector  $d_j = [Z_{score}(t_{1j}), Z_{score}(t_{2j}), \dots, Z_{score}(t_{mj})]$  for a particular value of  $j$ . Having this representation for each document an author profile  $A_k$  can then be created by taking the mean of these vectors for all the articles known to be written by that particular author.

Next we represent the query text  $Q$  in the same manner, as a vector of Z-scores. We then find the author profile that is closest to  $Q$  using equation 1, the distance function being defined as below.

$$D_1(Q, A_j) = \frac{1}{T} \cdot \sum_{i=1}^T |Z_{score}(t_{iq}) - Z_{score}(t_{ij})|$$

$t_{iq}$  denotes term  $t_i$  in query text, and  $t_{ij}$  denotes term  $t_i$  in author profile  $j$ .

### 3.2 Chi-Squared distance based method

Chi-Squared distance based method is based on the well known Pearson's  $\chi^2$  test, used to compute the similarity between two distributions. In this method a document is represented as a vector  $d_j = [p(t_{1j}), p(t_{2j}), \dots, p(t_{mj})]$ , where  $p(t_{ij})$  is normalised frequency of term  $t_i$  in a given document  $j$ . Author profile  $A_k$  is prepared by first combining all the documents pertaining to a particular author  $k$ , and then calculating the normalised frequency for this combined document. Considering  $Q$  and  $A_k$  as term distributions we can now use  $\chi^2$  distance to find the degree of similarity between the two. The distance function in this case is as shown below

$$D_2(Q, A_k) = \sum_{i=1}^T \frac{(q(t_i) - a_k(t_i))^2}{a_k(t_i)}$$

where  $q(t_i)$  is the normalised frequency for term  $t_i$  the query text  $Q$  and  $a_k(t_i)$  is that for the  $k^{th}$  author profile.

### 3.3 Z-Score based method

This method is currently the state of the art method for authorship attribution using quantitative analysis. It was proposed by Savoy (2012) and is a modification of the Delta method mentioned in section 3.1. One of the two major modifications is the method of calculating Z-Score. Savoy (2012) proposed using the expected value of term frequency and the expected standard deviation compared to the sample mean and sample standard deviation that were used in Delta method. So in this case any term  $t_{ij}$ ,  $i^{th}$  term in  $j^{th}$  document, is considered to be drawn from a binomial distribution with parameters  $n_0$  and  $p(t_i)$ .  $n_0$  is the length of the document for which  $t_{ij}$  is to be estimated and  $p(t_i)$  is the probability of term  $t_i$  occurring in the entire corpus. Hence the expected value for  $t_{ij}$  is  $n_0.p(t_i)$  and the expected standard deviation is  $\sqrt{n_0.p(t_i).(1 - p(t_i))}$ . The modified Z-score can then be calculated as

$$Z_{score}^*(t_{ij}) = \frac{tf_{ij} - n_0.p(t_i)}{\sqrt{n_0.p(t_i).(1 - p(t_i))}} \quad (3)$$

This Z-Score can then be used in the same way as used in Delta method. Another change in this method as compared to the Delta method is the distance function used.

$$D_3(Q, A_j) = \frac{1}{T} \cdot \sum_{i=1}^T \left( Z_{score}^*(t_{iq}) - Z_{score}^*(t_{ij}) \right)^2$$

where  $t_{iq}$  denotes term  $t_i$  in query text, and  $t_{ij}$  denotes term  $t_i$  in author profile  $j$ .

### 3.4 K.L.D. based method

K.L.D. based method is somewhat similar to the Chi-squared distance method in that this method also looks upon normalised word frequencies as a probability distribution. The author profiles and document profiles in this case are exactly the same as that in the Chi-squared distance based method. Kullback Leibler Divergence between the two probability distributions, namely author profile  $A_k$  and query text  $Q$  is defined as below

$$D_{KL}(Q||A_k) = \sum_{i=1}^T \ln \left( \frac{a_k(t_i)}{q(t_i)} \right) q(t_i)$$

where  $q(t_i)$  is the normalised frequency for term  $t_i$  the query text  $Q$  and  $a_k(t_i)$  is that for the  $k^{th}$  author profile. Author with profile  $A_k$  with minimum divergence from  $Q$  is considered to be the author for the disputed text.

## 4 Results and Evaluation

In this section we present the results of applying these four aforementioned techniques on our corpus. We also include one more technique apart from these four in which we use Delta method albeit with distance function  $D_3$ . We use the same evaluation strategy used by Savoy (2012). At a time we choose one article to be the disputed text  $Q$  and use all other articles to create the author profiles  $A_k$ . This is repeated for every article present in the corpus. Accuracy is then calculated in two ways: by finding the total number of articles attributed correctly irrespective of the authors (micro average) and by finding the accuracy for each author individually and then defining the overall accuracy as the average of these individual values (macro average). While experimenting with the number of texts required to create an author profile, for each article we select  $p$  articles from each author to create the author profiles. The concept of macro and micro average remain the same. But since we are selecting these  $p$  articles randomly, we perform a 10-fold cross validation to assure statistically significant results. In this case we report mean accuracy. Table 1 below shows the result for using different values of  $\lambda$ , with  $X$  and  $N$  remaining constant. All the terms with  $tf > 10$  and  $df > 2$ , were considered for the  $Z_{score}$  and  $K.L.D.$  based approaches while for Delta method top 400 terms were considered. For the chi-square based method the condition  $tf > 2$  was used. All these conditions are based on the best performing parameter value as found by (Savoy, 2012) and hence would make a good starting point. Above this we consider only those terms that belong to at least two author profiles so as not to make the task trivial. The size of the training set for this experiment was  $N = N_{max}$  i.e. all the available articles (except the query text  $Q$ ) are used to create the author profile. For each experiment the best performing parameter value is considered to be the baseline and other values are compared against them for statistically significant difference, using a two sided sign test.

Method	Parameter	Micro-Average	Macro-Average
Z-Score	$\lambda = 0$	86.14%	87.38% <sup>†</sup>
	$\lambda = 0.1$	<b>88.73%</b>	<b>90.45%</b>
Delta ( $D_1$ )	$\lambda = 0$	26.10% <sup>†</sup>	24.69% <sup>†</sup>
Delta ( $D_3$ )	$\lambda = 0$	84.24% <sup>†</sup>	86.00% <sup>†</sup>
KLD	$\lambda = 0.01$	77.17% <sup>†</sup>	70.38% <sup>†</sup>
	$\lambda = 0.001$	88.57%	85.44% <sup>†</sup>
$\chi^2$ Method	$\lambda = 0$	12.15% <sup>†</sup>	14.73% <sup>†</sup>

Table 1: Effect of variation in  $\lambda$

<sup>†</sup> Significant performance difference ( $\alpha = 1\%$ , two-sided sign test)

For further experiments we consider only the best performing value of the smoothing parameter and show that with proper feature selection, *i.e.* by selecting proper number of terms, K.L.D. based approach can give results comparable to the state of the art Z-score based approach. Chi-squared method and Delta method (using  $D_1$  distance) perform poorly and hence we do not consider them in further experimentation. All further experiments are performed only on Z-Score based method, Delta method (using  $D_3$  distance) and K.L.D. based method.

Method	Parameter	Micro-Avg	Macro-Avg
Z-Score	$tf > 10, df > 3$	88.73%	90.45% <sup>†</sup>
	$tf > 100, df > 3$	84.33% <sup>†</sup>	86.45% <sup>†</sup>
Delta ( $D_3$ )	Top 100 terms	76.10% <sup>†</sup>	74.69% <sup>†</sup>
	Top 400 terms	84.24% <sup>†</sup>	86.00% <sup>†</sup>
KLD	$tf > 10, df > 3$	88.57% <sup>†</sup>	85.44% <sup>†</sup>
	$tf > 100, df > 3$	90.55%	88.75%
	$tf > 1000, df > 3$	<b>91.35%</b>	<b>91.73%</b>

Table 2: Effect of variation in  $X$

<sup>†</sup> Significant performance difference ( $\alpha = 1\%$ , two-sided sign test)

Table 2 shows the variation in performance of these methods when the number of terms are varied. For Z-score based method and K.L.D. based method we choose terms based on their term frequencies in the corpus. We keep document frequency constant because increasing it would lead to selection of only those terms which are prevalent across more number of documents. These terms will make the author profiles less distinguishable and result in poor overall performance. For Delta method fewer number of terms always perform better (Burrows, 2002). Hence we use 100 and 400 terms respectively as done by (Burrows, 2002) and followed by (Savoy, 2012)

Further we investigate the effect of reducing the training set *i.e.* the number of texts used to create author profile. For this we select the smoothing parameter and the number of terms that performed best in the previous two experiments. For Z-Score based method we use the criteria  $tf > 10, df > 3$ , for K.L.D. based method we use  $tf > 1000, df > 3$  and for delta method we use top 400 most frequent terms. Table 3 shows the performance of the three systems as we vary the size of training set.  $N_{max}$  refers to the maximum number of articles that can be used to create the author profiles. In our case it is  $N_k - 1$ , where  $N_k$  is the total number of documents for the  $K^{th}$  author. Clearly when the size of the training set is small K.L.D. based method fares much better than all other techniques.

Method	Parameter	Micro-Average	Macro-Average
Z-Score	$N = 10$	52.14% <sup>†</sup>	54.17% <sup>†</sup>
	$N = 40$	82.39% <sup>†</sup>	84.45% <sup>†</sup>
	$N = N_{max}$	88.73%	90.45%
Delta	$N = 10$	22.10% <sup>†</sup>	23.69% <sup>†</sup>
	$N = 40$	64.14% <sup>†</sup>	65.50% <sup>†</sup>
	$N = N_{max}$	84.24% <sup>†</sup>	86.00% <sup>†</sup>
KLD	$N = 10$	72.35% <sup>†</sup>	75.34% <sup>†</sup>
	$N = 40$	90.25%	91.03%
	$N = N_{max}$	<b>91.35%</b>	<b>91.73%</b>

Table 3: Effect of variation in  $N$

<sup>†</sup> Significant performance difference ( $\alpha = 1\%$ , two-sided sign test)

## 5 Conclusion

Looking at the above results we can conclude that for Gujarati newspaper articles K.L.D. based authorship attribution with proper parameter selection is comparable to the current state of art Z-score based method when sufficient number of articles are available as a training set. But when the number of training examples are less then K.L.D. based method outperforms the Z-score based method. This might be because by normalising each of the terms' frequency,  $Z_{score}$  effectively considers each term to be of same importance. This might not be true as the distribution of terms that occur in most of the documents should ideally be a better signature as compared to the terms that occur in only a few documents of the author. *K.L.D.* inherently takes into account the occurrence frequency by weighting each term with the probability of its occurrence and hence performs better.

## Acknowledgement

This research is supported by part by the *Cross Lingual Information Access* project funded by *D.I.T., Government of India*.

## References

- M Bagavandas and G Manimannan. 2008. Style consistency and authorship attribution: A statistical investigation\*. *Journal of Quantitative Linguistics*, 15(1):100–110.
- John Burrows. 2002. Delta: A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17(3):267–287.
- David I Holmes. 1994. Authorship attribution. *Computers and the Humanities*, 28(2):87–106.
- Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. 2002. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412.
- Frederick Mosteller and David Wallace. 1964. Inference and disputed authorship: The federalist.
- Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj. 2003. Language independent authorship attribution using character level language models. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 267–274. Association for Computational Linguistics.
- Jacques Savoy. 2012. Authorship attribution based on specific vocabulary. *ACM Transactions on Information Systems (TOIS)*, 30(2):12.
- George Kingsley Zipf. 1932. Selected studies of the principle of relative frequency in language.

# Modeling User Leniency and Product Popularity for Sentiment Classification

Wenliang Gao\*, Naoki Yoshinaga<sup>†</sup>, Nobuhiro Kaji<sup>†</sup> and Masaru Kitsuregawa<sup>†‡</sup>

\*Graduate School of Information Science and Technology, The University of Tokyo

<sup>†</sup>Institute of Industrial Science, The University of Tokyo

<sup>‡</sup>National Institute of Informatics

{wl-gao, ynaga, kaji, kitsure}@tkl.iis.u-tokyo.ac.jp

## Abstract

Classical approaches to sentiment classification exploit only textual features in a given review and are not aware of the personality of the user or the public sentiment toward the target product. In this paper, we propose a model that can accurately estimate the sentiment polarity by referring to the *user leniency* and *product popularity* computed during testing. For decoding with this model, we adopt an approximate strategy called “two-stage decoding.” Preliminary experimental results on two real-world datasets show that our method significantly improves classification accuracy over existing state-of-the-art methods.

## 1 Introduction

Document-level sentiment classification estimates the sentiment polarity for a given subjective text (hereafter, review). Traditionally, researchers have tried to estimate the sentiment polarity from only the textual content of the review (Pang and Lee, 2004; Li et al., 2011). However, since reviews are written by a user to express his/her emotion toward a particular product, taking the users and products into consideration would play an important role in solving this task.

Recently, the increase of opinionated text within social media, e.g., *Twitter*, has motivated researchers to exploit the user or product information in the sentiment classification task. Some researchers take advantages of the friend relation in a social network because friends are likely to hold common tastes (Tan et al., 2011; Seroussi et al., 2010; Speriosu et al., 2011). Others incorporate user- or product-specific  $n$ -gram features (Li et al., 2011; Seroussi et al., 2010). Although these studies have showed that user or product information is useful for sentiment classification, they implicitly

assume that the same users or products appear in both training and testing data. Thus, to train such a model, a large amount of the reviews should be labeled for each user and each product. In a real-world scenario, however, this is unrealistic since new users and products are ceaselessly emerging and labeling reviews written by such users (or on such products) is impractical.

In the real world, different users have different rating standards, while different products receive different rating tendencies. For example, a critical person is likely to point out flaws and gives negative ratings, while a popular product receives more praise than negative feedback. We refer to these user- or product-specific polarity biases as *user leniency* and *product popularity*, respectively. A sentiment classifier would resort to these biases when textual features are not reliable enough to estimate the sentiment polarity.

In this study, we build a model that automatically computes and uses *user leniency* and *product popularity* for sentiment classification. We represent these biases with two types of real-valued global features. Because these features and the labels of the test reviews mutually depend on each other, it is challenging to globally optimize a configuration of polarity labels for a given set of reviews. We here adopt a two-stage decoding strategy (Krishnan and Manning, 2006) for resolving the mutual dependencies in our model.

We evaluated our method on two real-world datasets (Blitzer et al., 2007; Maas et al., 2011). Experimental results demonstrated that the proposed method significantly improved the classification accuracy against the state-of-the-art methods (Dredze et al., 2008; Seroussi et al., 2010).

The remainder of this paper is organized as follows. We first discuss some related work in Section 2. We describe our method in Section 3. We then report experimental results in Section 4. Finally, we conclude our study in Section 5.



## 2 Related Work

Recently, social media such as *Twitter* has attracted much attention from researchers because it is now apparently the major source of subjective text on the Web. The traditional text-based methods, such as Pang *et al.* (2002), could not easily handle such short and informal text (Jiang *et al.*, 2011).

Tan *et al.* (2010) and Speriosu *et al.* (2011) exploited the user network behind a social media website (*Twitter* in their case) and assumed that friends give similar ratings towards similar products. Seroussi *et al.* (2010) proposed a framework that computes users' similarity on the basis of their usage of text and their rating histories. They then classify a given review by referring to ratings given for the same product by other users who are similar to the user in question. However, such user networks are not always available in the real world.

Li *et al.* (2011) incorporate user- or product-dependent  $n$ -gram features into a classifier. They argue that users use a personalized language to express their sentiment, while the sentiment toward a product is described by product-specific language. This approach, however, requires the training data to contain reviews written by test users and written for test products. This is infeasible since labeling reviews requires too much manual work.

## 3 Method

Given a set of reviews,  $\mathcal{R}$ , our task is to estimate label  $y_r \in \{+1, -1\}$  for each review,  $r \in \mathcal{R}$ , with estimation function  $g(\mathbf{x}_r)$ :

$$g(\mathbf{x}_r) = \mathbf{w}^T \mathbf{x}_r, \quad (1)$$

$$y_r = \begin{cases} +1 & \text{if } g(\mathbf{x}_r) > 0 \\ -1 & \text{otherwise} \end{cases},$$

where  $\mathbf{x}_r$  is  $r$ 's feature vector and  $\mathbf{w}$  is the weight vector.

### 3.1 Idea

Our interest is to exploit user leniency and product popularity to improve sentiment classification. We encode each of them into two real-valued global features, which are detailed in Section 3.2. Since these global features depend on the labels of the input reviews, we cannot independently estimate the labels of reviews. We then discuss a decoding strategy in Section 3.3.

Note that we assume to know which reviews are written by the same user and which are written on the same product. This assumption is realistic nowadays since user information is available in many real-world datasets (Blitzer *et al.*, 2007; Pang and Lee, 2004), while product information can be extracted from text if not available (Qiu *et al.*, 2011). We should emphasize here that our method does not require user profiles, product descriptions, or any sort of extrinsic knowledge on the users and products.

### 3.2 Features

The review  $r$ 's feature vector,  $\mathbf{x}_r$ , is composed of local features ( $\mathbf{x}_r^l$ ) and global features ( $\mathbf{x}_r^g$ ), such that  $\mathbf{x}_r = (\mathbf{x}_r^l, \mathbf{x}_r^g)$ . In this study, we use word  $n$ -grams ( $n = 1, 2$ ) in the textual content of the review as local features, while we encode the user leniency and product popularity into global features. We introduce four global features to capture the user leniency and product polarity:

$$\mathbf{x}_r^g = \{f_{-u}^+, f_{-u}^-, f_{-p}^+, f_{-p}^-\},$$

where the first two features,  $f_{-u}^+$  and  $f_{-u}^-$ , represent the user leniency as the ratio of positive and negative reviews written by the same user of  $r$ , while the other two features,  $f_{-p}^+$  and  $f_{-p}^-$ , represent the product popularity as the ratio of positive and negative reviews on the same product of  $r$ . The global features are thereby computed as:

$$f_{-u}^+(r) = \frac{|\{r_j \mid y_j = +1, r_j \in N_u(r)\}|}{|N_u(r)|},$$

$$f_{-u}^-(r) = \frac{|\{r_j \mid y_j = -1, r_j \in N_u(r)\}|}{|N_u(r)|},$$

$$f_{-p}^+(r) = \frac{|\{r_j \mid y_j = +1, r_j \in N_p(r)\}|}{|N_p(r)|},$$

$$f_{-p}^-(r) = \frac{|\{r_j \mid y_j = -1, r_j \in N_p(r)\}|}{|N_p(r)|},$$

where  $N_u(r)$  represents a set of reviews written by the same user as  $r$  and  $N_p(r)$  represents a set of reviews written for the same product as  $r$ , respectively:

$$N_u(r) = \{r' \mid u_r = u_{r'} \wedge r \neq r'\},$$

$$N_p(r) = \{r' \mid p_r = p_{r'} \wedge r \neq r'\}.$$

### 3.3 Decoding

Because global features are computed for each user or product, we want to process as many test

reviews at once so that they include many reviews for each user or on each product to compute reliable global features. However, because the possible ways of assigning labels to a given set of reviews,  $\mathcal{R}$ , is  $2^{|\mathcal{R}|}$  and the two types of global features introduce complex label dependencies to be resolved, exact decoding is computationally expensive even with dynamic programming. In this study, we thus resort to an approximate decoding strategy called “two-stage decoding” (Krishnan and Manning, 2006). It splits the decoding process into a local decoding stage and a global decoding stage. Each stage takes linear time with respect to the number of reviews processed. This strategy is thereby scalable to a larger number of test reviews.

At the first stage, all the global features are set to 0, and only local features are used to classify the reviews. In the second stage, labels estimated in the first stage are used to compute the values of the global features. The labels are then revised by using both local and global features. In our case, the two-stage decoding at first uses only word  $n$ -gram features to estimate the labels of reviews. The estimated labels are used to compute user leniency features and product popularity features. Then, the decoding revises the labels considering both the word  $n$ -gram features and the user leniency and product popularity features.

### 3.4 Training

We train a binary classifier as the score estimation function in Eq. 1, considering word  $n$ -gram features, user leniency features, and product popularity features. The values of global features are computed by using the gold labels. We assume that a value of the user leniency feature or product popularity feature for a review whose user has no other reviews or whose product has no other reviews is set to 0.

## 4 Experiments

We evaluated our method in terms of accuracy on two real-world datasets (Blitzer et al., 2007; Maas et al., 2011) for a document-level sentiment classification task.

For each review, we at first use OpenNLP<sup>1</sup> to detect sentence boundaries and tokenize each sentence in order to obtain word  $n$ -gram features. Following Pang *et al.* (2002)’s settings, we take nega-

<sup>1</sup><http://opennlp.apache.org/>

Dataset	Blitzer	Maas
No. of reviews	188,350	50,000
No. of users	123,584	n/a
No. of products	101,021	7,036
No. of reviews/user	1.5	n/a
No. of reviews/products	1.9	7.1

Table 1: Dataset statistics.

tion (such as *n’t* and *cannot*) into consideration. Because features with low frequency are unreliable, any  $n$ -gram that appears less than six times in the training data are ignored.

We adopted a confidence weighted linear classifier (Dredze et al., 2008) as our binary classifier. This is because it has been reported to perform best on the sentiment classification task (Dredze et al., 2008).

### 4.1 Datasets

We used two datasets that were developed by Blitzer *et al.* (2007) and Maas *et al.* (2011). The datasets contain user/product and only product information. The statistics of the two datasets are summarized in Table 1.

The original Blitzer dataset contains more than 780,000 reviews (88% positive, 12% negative), which were collected from amazon.com across several domains, such as books, movies and games. We automatically delete reviews written by the same user on the same product, which results in about 740,000 reviews. Then, the reviews are balanced for positive and negative labels (94,175 reviews for each) to maintain consistency with the setting in other existing works.

The Maas dataset has 25,000 positive and 25,000 negative reviews on movies. The dataset provides a URL for each review, which represents the sentiment target, a movie. We thus use the URL as a unique identifier for the movie. The user information cannot be fully recovered, so we only model the product dependency on this dataset.

Our method performs best when the reviews written by/on the same user/product are in the same set (training or testing) since we can compute more reliable global features when we have more reviews written by/on the same user/product. In the two datasets, reviews were originally ordered by user or product. To prevent a seemingly unfair accuracy gain under this particular splitting, we randomly shuffled the reviews and performed

Method	Accuracy (%)	
	Blitzer	Maas
Seroussi <i>et al.</i> (2010)	89.37	n/a
Maas <i>et al.</i> (2011)	n/a	88.89 <sup>3</sup>
baseline	90.11	91.35
proposed	91.01 <sup>&gt;</sup>	92.68 <sup>&gt;&gt;</sup>

Table 2: Accuracy on review datasets. Accuracy marked with “>>” or “>” was significantly better than baseline ( $p < 0.01$  or  $0.01 \leq p < 0.05$  assessed by McNemar’s test).

a 2-fold cross-validation.

## 4.2 Results

In this section, we report the accuracy of our sentiment classifier. Accuracy is measured as the number of correctly classified reviews divided by the number of all the reviews. We prepared two baseline classifiers to see the advantage of our classifier. As one baseline, we used a confidence-weighted linear classifier (Dredze *et al.*, 2008) that takes only textual features into account. As another baseline, we implemented a user similarity-based method proposed by Seroussi *et al.* (2010).<sup>2</sup> The similarity of users is computed by using a word  $n$ -gram Jaccard distance (called “AIT” in Seroussi *et al.* (2010)). When the user of an input review is unseen in the training data, a default classifier, which is trained with all the training reviews, is used to classify the review.

Table 2 shows the experimental results. The proposed method significantly improved the classification accuracies across the two datasets. A larger improvement was acquired on the Maas dataset because the average number of reviews for each product in the dataset was larger than that in the Blitzer dataset.

**Impact of size on test reviews** In our method, since global features play a key role, acquiring

<sup>2</sup>We built user-specific classifiers for users who wrote reviews with positive polarity and negative polarity more than a pre-specified threshold. After several trials, the threshold was set to be 5 to gain the best performance.

<sup>3</sup>This result was computed under a different splitting from ours. Under Maas *et al.* (2011)’s splitting, the accuracy for the baseline and proposed method was 90.83% and 92.29%. The main difference between our baseline and their method is the features. They use only unigram features, while we use unigram and bigram as features. Using only unigram features under their splitting, the accuracy of the baseline method was 87.8%.

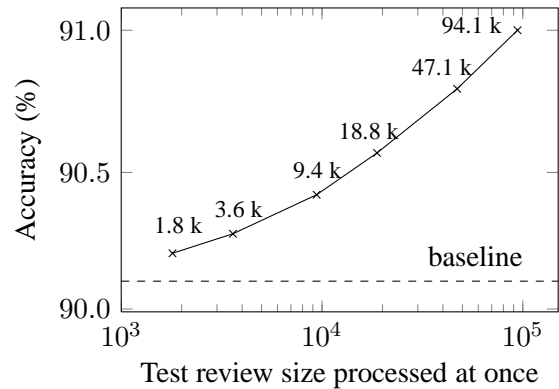


Figure 1: Accuracy when we changed the size of test reviews processed at once by our classifier.

more reliable global features is our major concern to make the improvement more significant.

We thus performed 2-fold cross-validation with the same splitting for the Blitzer dataset, while changing the size of test reviews processed at once to investigate the impact of test review size on classification accuracy. In this experiment, we split the test reviews into equal-sized smaller subsets and applied our classifier independently to each of the subsets.

As shown in Figure 1, when we processed a larger number of test reviews at once, the accuracy increased. This result confirms our expectations.

## 5 Conclusion

We presented a model that captures and uses user leniency and product popularity for sentiment classification. Different from the previous studies that are aware of the user and product of the review, our model does not require the training data to contain reviews written by the test users or written on the test products. To infer labels under our proposed model, we investigated a two-stage decoding strategy.

We conducted experiments on two real-world datasets to demonstrate the effectiveness of our proposed method. The method performed more accurately than did the baseline method, which only uses  $n$ -gram features, and an existing user-aware approach. We also showed that processing more test reviews at once lead to better accuracy.

We plan to publish our code and datasets.<sup>4</sup> A detailed exploration of this work will be reported in Gao *et al.* (2013).

<sup>4</sup><http://www.tkl.iis.u-tokyo.ac.jp/~wl-gao/>

## References

- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, pages 440–447, Prague, Czech Republic.
- Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of ICML*, pages 264–271, New York, NY, USA.
- Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Collective sentiment classification based on user leniency and product popularity. In *Proceedings of PACLIC*, Taipei, Taiwan. to appear.
- Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings ACL-HLT*, pages 151–160, Portland, Oregon, USA.
- Vijay Krishnan and Christopher D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of COLING-ACL*, pages 1121–1128, Sydney, Australia.
- Fangtao Li, Nathan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. 2011. Incorporating reviewer and product information for review rating prediction. In *Proceedings of IJCAI*, pages 1820–1825, Barcelona, Catalonia, Spain.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL-HLT*, pages 142–150, Portland, Oregon, USA.
- Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, pages 271–278, Stroudsburg, PA, USA.
- Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.
- Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2010. Collaborative inference of sentiments from texts. In *Proceedings of UMAP*, pages 195–206, Berlin, Heidelberg.
- Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of EMNLP, workshop on Unsupervised Learning in NLP*, pages 53–63, Edinburgh, UK.
- Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of KDD*, pages 1397–1405, New York, USA.

# A Generalized LCS Algorithm and Its Application to Corpus Alignment

Jin-Dong Kim

Database Center for Life Science (DBCLS), ROIS, Japan  
jdkim@dbcls.rois.ac.jp

## Abstract

The paper addresses the problem of text variation which often hinders interoperable use or reuse of corpora and annotations. A systematic solution is presented based on a variation of Longest Common Sequence algorithm. An empirical experiment with 20 full text articles shows it works well with a real world application.

## 1 Introduction

Corpus with annotation is regarded indispensable for the development of natural language processing (NLP) technology. As so, many corpora with annotation have been developed, and many groups are working with new annotation projects.

As various annotated corpora accumulate in the field, reusability and interoperability is becoming an important issue (Cohen et al., 2005; Johnson et al., 2007; Campos et al., 2012). Among others, Wang et al. (2010) reports that there are a number of corpora that claim to have annotations for protein or gene names, e.g. Genia (Kim et al., 2003), Aimeid (Bunescu et al., 2004), and Yapex (Franzén et al., 2002), and that, however, the protein annotations in those corpora are substantially different to each other, which calls for an interoperable interpretation of the annotations for integrative reuse of them. Rebholz-Schuhmann et al. (2011) investigates aggregation of variable named entity annotations in large scale, which also show the importance of interoperable use of corpus annotation.

There also have been efforts for the interoperability of corpora and annotations from a perspective of encoding and representation, e.g., Linguistic Annotation Framework (LAF) (Ide and Romary, 2004) and Open Linguistics<sup>1</sup>. Without a doubt, those efforts contribute to improving the interoperability of corpora and annotation.

<sup>1</sup><http://linguistics.okfn.org/>

A.

T	G	F	-	b	e	t	a		a	c	t	s	...
0	1	2	3	4	5	6	7	8	9	10	11	12	

(0, 8), Protein

B.

T	G	F	-	&	b	e	t	a	;		a	c	t	s	...
0	1	2	3	4	5	6	7	8	9	10	11	12			

(0, 10), Signaling\_molecule

C.

T	G	F	-	$\beta$		a	c	t	s	...
0	1	2	3	4	5	6	7	8	9	

(0, 5), Protein, Signaling\_molecule

Figure 1: Text variations and annotation to them

This paper addresses another type of problem, *text variation*, which often hinders interoperable use or reuse of corpora and annotations in real world applications. As far as the author knows, it is the first attempt to develop a definite and systematic solution to the problem.

The problem of text variant is explained in detail in Section 2, while the solutions are presented in Section 3 and 4 After discussions on its real world application in Section 5, the paper is concluded in Section 6.

## 2 Task definition

Figure 1 illustrates a simple example of the problem to be addressed: A, B and C are text variants from the same document; The position index of the equivalent text spans, “TGF-beta”, “TGF-&beta;”, and “TGF- $\beta$ ”, are different to each other; And, the annotations made to the spans are not directly interoperable, although they are made to conceptually the same span of the same document.

Note that the example is extremely simplified for the ease of understanding. In reality, the problem is much more complex: a text, as the tar-

get of annotation, is often as long as hundreds, or thousands of characters, or even much longer, and a single local variant affects the entire remaining portion of the text, in a cumulative way.

Nowadays, the widespread use of Unicode is one of the reasons of text variant, particularly when it comes to text processing, because many NLP tools, e.g., syntactic parsers, cannot handle Unicode characters properly. Thus, during many annotation projects, Unicode characters, e.g., Greek letters, are spelled out into ASCII alphabets, like *beta* in Figure 1 A. Sometimes, extra symbols, e.g., ‘&’ and ‘;’, are inserted to delimit Unicode-origin sequences, like in B.

Suppose that two independent annotation projects took the text of C into their corpora, and their preprocessors spelled out Greek letters differently like in A and B. The projects may produce different annotations according to their interest and perspectives. While those annotations may serve their goals individually, further benefit, e.g., reuse, comparison, or aggregation, can be gained from interoperable use of them. In the example, we want the annotations,  $(0, 8, \text{Protein})^2$  from A and  $(0, 10, \text{Signaling\_molecule})$  from B, to be transferable to C, or to each other. However, the variation of text poses a challenge: we need to compute the mapping between variations of text.

For standoff annotation, a text defines a one-dimensional Cartesian coordinate system, whereon any position on the text is specified. We thus cast the problem to the task of finding a mapping function from a one-dimensional Cartesian coordinate system to another, when they are filled with comparable values (characters). In Figure 2,  $\delta_{A \rightarrow C}$  is a mapping function from A to B, which enables transferring the annotation to the source text.  $\delta_{C \rightarrow A}$  is the mapping for the opposite direction. Once those functions are obtained, any annotation produced by the project A can be transferred to the original text, and vice versa.

### 3 LCS for text mapping

For most cases, text mapping can be computed using Longest Common Sequences (LCS) algorithms (Bergroth et al., 2000). LCS is a well

<sup>2</sup>Throughout the paper, we make the span specification in the style of BioNLP shared task (Kim et al., 2009), where the beginning of a span is specified by the number of characters preceding the span, and the end by the number of characters up to the end of the span.

0	1	2	3	4	5	6	7	8	9	10	11	...
0	1	2	3	4	4	4	4	5	6	7	8	

0	1	2	3	4	5	6	7	8	9	...
0	1	2	3	4	8	9	10	11	12	

Figure 2: Mapping between text variations

		T	G	F	-	b	e	t	a		a	c	t	s
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	<b>1</b>	1	1	1	1	1	1	1	1	1	1	1	1
G	0	1	<b>2</b>	2	2	2	2	2	2	2	2	2	2	2
F	0	1	2	<b>3</b>	3	3	3	3	3	3	3	3	3	3
-	0	1	2	3	<b>4</b>	4	4	4	4	4	4	4	4	4
$\beta$	0	1	2	3	4	<b>4</b>	4	4	4	4	4	4	4	4
	0	1	2	3	4	4	4	4	4	<b>5</b>	5	5	5	5
a	0	1	2	3	4	4	4	4	4	5	<b>6</b>	6	6	6
c	0	1	2	3	4	4	4	4	4	5	6	<b>7</b>	7	7
t	0	1	2	3	4	4	4	4	4	5	6	7	<b>8</b>	8
s	0	1	2	3	4	4	4	4	4	5	6	7	8	<b>9</b>

Figure 3: The LCS table for “TGF- $\beta$  acts” and “TGF-beta acts”. The place of text variation is indicated in gray

known problem, based on which the UNIX command, *diff*, is implemented. Algorithm 1 is a dynamic algorithm to compute the length of LCS of any two strings. Figure 3 shows the resulted LCS table for example strings. From the LCS table, the *diff* between the two strings - see Figure 4<sup>3</sup> - can be read out by Algorithm 2.

---

#### Algorithm 1 LCS computation

---

```

1: function LCS( $X[1..m], Y[1..n]$ )
2:    $C = \text{ARRAY}(0..m, 0..n)$ 
3:   for  $i := 0..m$  do
4:      $C[i, 0] := 0$ 
5:   end for
6:   for  $j := 0..n$  do
7:      $C[0, j] := 0$ 
8:   end for
9:   for  $i := 1..m$  do
10:    for  $j := 1..n$  do
11:      if  $X[i] = Y[j]$  then
12:         $C[i, j] := C[i - 1, j - 1] + 1$ 
13:      else
14:         $C[i, j] := \text{MAX}(C[i, j - 1], C[i - 1, j])$ 
15:      end if
16:    end for
17:  end for
18: end function

```

---

Algorithm 1 has time and space complexities of  $O(mn)$ , where  $m$  and  $n$  are the length of the strings. For many real world applications, Hunt-

<sup>3</sup>In the first column, the minus (‘-’) and plus (‘+’) signs indicate *deletion* and *insertion* operations, respectively.

---

**Algorithm 2** Reading out *diff* from LCS table
 

---

```

1: D = STACK
2: i := m
3: j := n
4: while i ≠ 0 or j ≠ 0 do
5:   if i > 0 and j > 0 and X[i] = Y[j] then
6:     PUSH(D, ['=', X[i], Y[j]])
7:     i := i - 1
8:     j := j - 1
9:   else if j > 0 and (i = 0 or C[i, j-1] > C[i-1, j]) then
10:    PUSH(D, ['+', nil, Y[j]])
11:    j := j - 1
12:   else
13:    PUSH(D, ['-', X[i], nil])
14:    i := i - 1
15:   end if
16: end while

```

---

McIlroy algorithm (Hunt and McIlroy, 1976) is frequently used, which regularly beats the complexities of the dynamic algorithm with typical inputs. Once the *diff* in Figure 4 is obtained, getting the mapping  $\delta_{C \rightarrow A}$  is straightforward.

=	0	T	0	T
=	1	G	1	G
=	2	F	2	F
=	3	-	3	-
-	4	$\beta$		
+			4	b
+			5	e
+			6	t
+			7	a
=	5		8	
=	6	a	9	a
=	7	c	10	c
=	8	t	11	t
=	9	s	12	s

Figure 4:  $\text{diff}(C, A)$

The LCS algorithm works fine when text variations occur only in isolation individually as in Figure 4. Sometimes, however, text variations occur successively, causing what we call the *successive variation* problem. It is illustrated in the *diff* result in Figure 5, where two Unicode characters, ‘ $\beta$ ’ and ‘ $\text{^-}$ ’<sup>4</sup> appear successively. The source position, 5, needs to be precisely mapped to the target position, 8, which however the LCS-diff algorithms cannot find: while the mapping,  $\delta(4) \rightarrow 4$ , is obvious, there is no clue as to which position, among 5, 6, 7 and 8, the next one,  $\delta(5)$  to be mapped to.

#### 4 Generalized LCS algorithm

To address the problem of *successive variations*, we need to inform the algorithm of equivalent sequences, e.g.,  $\beta$  and *beta*. We call a collection of

<sup>4</sup>long hyphen in Unicode

=	0	T	0	T
=	1	G	1	G
=	2	F	2	F
=	3	-	3	-
-	4	$\beta$		
-	5	-		
+			4	b
+			5	e
+			6	t
+			7	a
+			8	-
=	6	i	9	i
=	7	n	10	n
=	8	d	11	d
=	9	u	12	u
=	10	c	13	c
=	11	e	14	e
=	12	d	15	d

Figure 5: The result of LCS-Diff for “*TGF- $\beta$ -induced*” and “*TGF-beta-induced*”

equivalent sequences a *dictionary*, and modify Algorithm 1 as follows:

```

1: function GLCS(X[1..m], Y[1..n], D)
   ...
11-1: a, b := S(X[1..i], Y[1..j], D)
11-2: if a > 0 then
12:   C[i, j] := C[i-a, j-b] + 1

```

As indicated in line 1, it is invoked with a dictionary, *D*, which is a list of equivalent sequences, e.g., ( $\alpha$ , *alpha*), ( $\beta$ , *beta*), and so on. The 11’t line of Algorithm 1, which performs the comparison of the last characters, is modified to perform a general suffix comparison. The suffix comparison function, *S*, first performs the last-character-comparison for trivial cases, and performs a suffix comparison in variable length when the character comparison fails. The suffix comparison relies on the dictionary, *D*: if the two strings have matching suffixes in the end according to the dictionary, it returns the length of the suffixes, which is received by *a* and *b* in the modified algorithm.

Following is the modification to Algorithm 2:

```

5-1: a, b := S(X[a..i], Y[b..j])
5-2: if i > 0 and j > 0 and a > 0 then
6-1:   if a = b = 1 and X[i] = Y[j] then
6-2:     push(D, ['=', X[i], Y[j]])
6-3:   else
6-4:     for p := i-a+1..i do
6-5:       push(D, ['-', X[p], nil])
6-6:     end for
6-7:     for q := j-b+1..j do
6-8:       push(D, ['+', nil, Y[q]])
6-9:     end for
6-10:  end if
7:   i := i - a
8:   j := j - b

```

Using it, the *diff* of the successive variation example is obtained as in Figure 6, where the mapping

=	0	T	0	T
=	1	G	1	G
=	2	F	2	F
=	3	-	3	-
+	4	$\beta$	4	b
+			5	e
+			6	t
+			7	a
-	5	-	8	-
+	6	i	9	i
=	7	n	10	n
=	8	d	11	d
=	9	u	12	u
=	10	c	13	c
=	11	e	14	e
=	12	d	15	d

Figure 6: The result of GLCS-Diff for “*TGF-beta-induced*” and “*TGF-beta-induced*”

of ‘-’ and ‘-’ is properly represented, solving the problem of successive variations.

As the modified algorithm generalizes the last-character-comparison to the variable-length-suffix-comparison, we call it a *generalized LCS (GLCS)* algorithm. With an empty dictionary, GLCS works exactly the same as LCS. Using a *suffix tree* algorithm, GLCS has the worst case time complexity,  $O(mnl)$ , where  $l$  is the length of the longest entry in the dictionary.

While the performance of GLCS relies on the dictionary, in fact, it works well even with an incomplete dictionary. For example, to get the result in Figure 6, having either ( $\beta$ , *beta*) or (-, -) in the dictionary is enough. This feature contributes to the robustness of GLCS in real world applications.

## 5 Application and evaluation

The proposed solution is implemented into PubAnnotation<sup>5</sup>, a storage system for corpora and annotations. The system is developed to share corpora and annotations developed by several annotation projects. The system maintains a collection of texts taken from a number of sources, e.g., PubMed<sup>6</sup> and PubMed Central<sup>7</sup>, and supplies them to the annotation projects. The annotations produced by the annotation projects are collected back to PubAnnotation for sharing.

As the annotation projects are conducted by different groups independently, when the resulted annotations are submitted to the storage system, the

<sup>5</sup><http://pubannotation.org>

<sup>6</sup><http://www.ncbi.nlm.nih.gov/pubmed>

<sup>7</sup><http://www.ncbi.nlm.nih.gov/pmc/>

base texts often have been varied from the original, due to, e.g., Unicode-ASCII conversion, tokenization, or accidental insertion or deletion of characters. PubAnnotation handles all the mapping and alignment by using the LCS and GLCS algorithms. For performance, LCS is implemented using Hunt-McIlroy algorithm, and GLCS is implemented as presented in this paper. While using LCS as default, GLCS is only invoked when successive variations are detected. Because successive variations seldom occur, the cost for running GLCS is negligible. Yet, securing a solution for successive variations is important. When experimented with 10 full papers with 54,938 words and 6,007 annotation instances, the text mapping and annotation alignment took less than 10 seconds.

The accuracy of mapping and alignment is thoroughly verified using 10 full text papers with 58,360 words and 7,315 span annotations. Two versions of dictionary for GLCS were prepared: (A) one for all the standard set of Unicode characters<sup>8</sup>, and (B) another only for the Unicode characters for whitespace and punctuation symbols. The system successfully aligned all the annotations even with the smaller one, (B). It indicates that when successive variations occur, in most cases, whitespace or punctuation symbols are mixed in it. At least it was the case in our application.

Another 10 full text papers with 54,369 words and 5,729 annotations were used for further verification. While keeping using the dictionary (B), the system is implemented to alert when an unsolvable case is detected. During the processing of the 10 papers, the alert was issued only once, which was caused by the Unicode sequence,  $\Delta\Delta$ . When the larger dictionary, (A), was used, the problem was not observed. So, it is true that the more complete the dictionary is, the higher the accuracy will be. The empirical results also suggest that, together with the alerting system, the proposed solution works reasonably well, even with minimal size of dictionary.

## 6 Conclusions

The solution presented in this paper is freely available as an open source Ruby library and also as a free service through the PubAnnotation storage system. We expect it to contribute to reduce the cost of community for interoperable use of corpora and annotations.

<sup>8</sup>As implemented in the standard *unicode* library.



## References

- L. Bergroth, H. Hakonen, and T. Raita. 2000. A survey of longest common subsequence algorithms. In *Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00)*, SPIRE '00, pages 39–, Washington, DC, USA. IEEE Computer Society.
- Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. 2004. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.
- David Campos, Sergio Matos, Ian Lewin, Jos Lus Oliveira, and Dietrich Rebholz-Schuhmann. 2012. Harmonization of gene/protein annotations: towards a gold standard medline. *Bioinformatics*, 28(9):1253–1261.
- K. Bretonnel Cohen, Philip V Ogren, Lynne Fox, and Lawrence Hunter. 2005. Empirical data on corpus design and usage in biomedical natural language processing. In *AMIA annual symposium proceedings*, pages 156–160.
- Kristofer Franzén, Gunnar Eriksson, Fredrik Olsson, Lars Asker, Per Lidén, and Joakim Cöster. 2002. Protein names and how to find them. *International Journal of Medical Informatics*, 67(13):49 – 61.
- James W. Hunt and M. Douglas McIlroy. 1976. An Algorithm for Differential File Comparison. Technical Report 41, Bell Laboratories Computing Science, July.
- Nancy Ide and Laurent Romary. 2004. International standard for a linguistic annotation framework. *Nat. Lang. Eng.*, 10(3-4):211–225, September.
- Helen Johnson, William Baumgartner, Martin Krallinger, K Bretonnel Cohen, and Lawrence Hunter. 2007. Corpus refactoring: a feasibility study. *Journal of Biomedical Discovery and Collaboration*, 2(1):4.
- Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182.
- Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop*, pages 1–9.
- Dietrich Rebholz-Schuhmann, Antonio Yepes, Chen Li, Senay Kafkas, Ian Lewin, Ning Kang, Peter Corbett, David Milward, Ekaterina Buyko, Elena Beisswanger, Kerstin Hornbostel, Alexandre Kouznetsov, Rene Witte, Jonas Laurila, Christopher Baker, Cheng-Ju Kuo, Simone Clematide, Fabio Rinaldi, Richard Farkas, Gyorgy Mora, Kazuo Hara, Laura I Furlong, Michael Rautschka, Mariana Neves, Alberto Pascual-Montano, Qi Wei, Nigel Collier, Md Chowdhury, Alberto Lavelli, Rafael Berlanga, Roser Morante, Vincent Van Asch, Walter Daelemans, Jose Marina, Erik van Mulligen, Jan Kors, and Udo Hahn. 2011. Assessment of ner solutions against the first and second calbc silver standard corpus. *Journal of Biomedical Semantics*, 2(Suppl 5):S11.
- Yue Wang, Jin-Dong Kim, Rune Sætre, Sampo Pyysalo, Tomoko Ohta, and Jun'ichi Tsujii. 2010. Improving the inter-corpora compatibility for protein annotations. *Journal of Bioinformatics and Computational Biology*, 8(5):901–916.

# Semantic Naïve Bayes Classifier for Document Classification

**How Jing, Yu Tsao**

Research Center for Information  
Technology Innovation  
Academia Sinica, Taipei, Taiwan  
{yu.tsao}@citi.sinica.edu.tw

**Kuan-Yu Chen, Hsin-Min Wang**

Institute of Information Science  
Academia Sinica, Taipei, Taiwan  
{kychen, whm}@iis.sinica.edu.tw

## Abstract

In this paper, we propose a semantic naïve Bayes classifier (SNBC) to improve the conventional naïve Bayes classifier (NBC) by incorporating “document-level” semantic information for document classification (DC). To capture the semantic information from each document, we develop semantic feature extraction and modeling algorithms. For semantic feature extraction, we first apply a log-Bilinear document modeling (LBDM) algorithm to transform each word into a semantic vector, and then apply principal component analysis (PCA) to the semantic space formed by the word vectors to extract a set of semantic features for each document. For semantic modeling, a semantic model is constructed using the semantic features of the training documents. In the testing phase, SNBC systematically integrates the semantic model and the conventional NBC to perform DC. The results of experiments on the 20 News-groups and WebKB datasets confirm that, with the semantic score, SNBC consistently outperforms NBC with various language modeling approaches.

## 1 Introduction

Document classification (DC) is an important task in the information retrieval (IR) and natural language processing (NLP) areas. In recent years, many approaches have been developed for DC. Among them, a category of approaches views DC as a traditional ranking problem. These approaches first represent a document with a feature vector, known as the vector space model (VSM), and then machine learning algorithms can be applied to accomplish classification. Notable examples belonging to this category include support vector machine (Joachims, 1998), decision tree (Comite *et al.*, 2003), logistic regression (Genkin *et al.*, 2005), and k-nearest neighbor (Kwon and Lee, 2003).

Another successful category of approaches is based on the naïve Bayes classifier (NBC). NBC assumes that a document is generated from a probabilistic model. In the offline training process, the model parameters are estimated from a set of training documents. When performing DC, NBC calculates a conditional probability  $P(c|d)$  (the posterior probability that document  $d$  belongs to class  $c$ ) and classifies the test document to the class that gives the highest  $P(c|d)$ . When calculating  $P(c|d)$ , NBC makes word independence (bag-of-words) assumptions and decomposes  $P(c|d)$  to a product of individual word probabilities. These word probabilities are usually characterized by a statistical language model. In practice, unigram language modeling (ULM) is a popular choice due to its effectiveness and computational efficiency (Peng and Schuurmans, 2003; Bai and Nie, 2004; Wu and Wang, 2012). However, since NBC with ULM only considers frequencies of words occurring in a class, it may suffer from the problems of data sparseness and word usage diversity, leading to performance degradations for DC. To deal with the data sparseness (zero probability) problem of ULM, many smoothing techniques, such as Laplace (Bai and Nie, 2004) and Jelinek-Mercer (Jelinek and Mercer, 1980) smoothing techniques have been developed.

The latent topic language modeling (LTM) approaches use a set of latent topics to decompose the relationships between documents and classes. Successful examples include latent semantic analysis (LSA) (Bellegarda, 2005; Deerwester *et al.*, 1990), probabilistic latent semantic analysis (PLSA) (Hofmann, 1999a; Hofmann, 1999b), and latent Dirichlet allocation (LDA) (Blei 2011; Griffiths and Steyvers, 2004; Blei *et al.*, 2003). For these approaches, classification scores are not computed directly based on the frequencies of the words but instead based on a set of latent topics along with the likelihoods that each class generates the respective topics. The use of latent topics effectively tackles the word usage diversity problem for ULM and performs DC in a concept matching manner.

Although NBC with LTM approaches have taken

the semantic information into account, the document-level semantic cues are not directly incorporated for the DC task (LTM approaches only extract the word frequency information from documents). In this paper, we intend to enhance the NBC-based approaches by incorporating document-level semantic information for DC. In the training phase, we estimate the parameters of the semantic model by using the training documents. In the testing phase, a semantic score is computed based on the given test document with a particular class. The final decision of DC is made based on a combined score from the semantic model and the traditional NBC. Since the proposed framework is derived based on the NBC framework, we name it “semantic NBC” (SNBC). We conduct experiments on two sets of corpora, namely 20 Newsgroups and WebKB. Experimental results demonstrate that SNBC consistently outperforms NBC with various language modeling approaches.

The remainder of this paper is organized as follows. Section 2 defines the notations and briefly reviews the related work. Section 3 introduces the proposed SNBC framework. Section 4 describes our experimental setup and discusses experimental results. Finally, we conclude this work in Section 5.

## 2 Related Work

In this section, we present notation and terminology to be used in the following discussions and review related work to the proposed SNBC framework.

### 2.1 Notation

The basic unit in a DC task is word, which is denoted as  $w$ , where  $w \in \mathbf{V}$ , and  $\mathbf{V}$  denotes the vocabulary set. A document is a sequence of words, and we denote a document by  $d$ , where  $|d|$  represents the total number of words in the document. A class is a predefined document class, and we denote a class by  $c$ . Assuming that we have  $|\mathbf{C}|$  distinct classes, the goal of DC is to classify a test document  $d_{test}$  into a specific class  $c$ , where  $c \in \mathbf{C}$ .

### 2.2 Naïve Bayes Classifier with Language Modeling

NBC performs DC in two stages: training and testing. In the training stage, a generative model is estimated based on the training documents for each class. In the testing stage, NBC calculates the posterior probability of each class given the evidence of the test document and selects the class that gives the highest probability

$$\hat{c} = \underset{c}{\operatorname{argmax}} P(c|d). \quad (1)$$

By applying Bayes' theorem on  $P(c|d)$ , we have

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \propto P(d|c)P(c), \quad (2)$$

where  $P(d|c)$  is the likelihood of document  $d$  under class  $c$ . Since NBC assumes that words in  $d$  are independent to each other,  $P(d|c)$  can be decomposed as

$$P(d|c) = \prod_{w \in d} P(w|c)^{n(w,d)}, \quad (3)$$

where  $P(w|c)$  is the class unigram probability, which indicates the likelihood of word  $w$  occurring in the class  $c$ , and  $n(w,d)$  denotes the frequency of word  $w$  occurring in  $d$ . Generally, a unigram language model (ULM) is used for calculating  $P(w|c)$ . However, ULM may encounter a data sparseness (zero-probability) problem. To deal with this problem, many smoothing techniques have been developed. Laplace and Jelinek-Mercer smoothing techniques are two successful examples.

The Laplace smoothing technique calculates  $P(w|c)$  by

$$P_{La}(w|c) = \frac{1+n(w,c)}{|\mathbf{V}| + \sum_{w \in \mathbf{V}} n(w,c)}, \quad (4)$$

where  $n(w,c)$  denotes the frequency of word  $w$  occurring in class  $c$ .

The Jelinek-Mercer smoothing technique calculates  $P(w|c)$  by

$$P_{JM}(w|c) = \lambda \frac{n(w,c)}{\sum_{w \in \mathbf{V}} n(w,c)} + (1-\lambda)P_{BG}(w), \quad (5)$$

where  $P_{BG}(w)$  is a background language model obtained from the entire training corpus. The tunable parameter  $\lambda$  in Eq. (5) can be determined based on a set of development data. Although many studies have proven that NBC with ULM can provide satisfactory performance, the method has a limitation: the classification process is based on literal term matching and only considers frequencies of words occurring in a class. Thus, NBC with ULM usually suffers from the issue of word usage diversity and polysemy, which can degrade the DC performance.

### 2.3 Latent Topic Modeling

In contrast to literal term matching, a latent topic language modeling (LTM) incorporates a set of latent topic variables to decompose the relationships between documents and classes. PLSA (Hofmann, 1999a; Hofmann, 1999b) and LDA (Blei, 2011; Griffiths and Steyvers, 2004; Blei *et al.*, 2003) are two representative LTM approaches. For PLSA,  $P(d|c)$  is formulated as

$$P_{\text{PLSA}}(d|c) = \prod_{w \in d} \left[ \sum_{k=1}^K P(w|T_k) P(T_k|c) \right]^{n(w,d)}, \quad (6)$$

where  $T_k$  is the  $k^{\text{th}}$  latent topic variable, and  $K$  denotes the total number of latent topics. The word-topic likelihood  $P(w|T_k)$  and topic-class likelihood  $P(T_k|c)$  can be estimated beforehand by maximizing the total log-likelihood of the training data. For traditional NBC with ULM, the model implicitly assumes that each word in a document is drawn from a single topic distribution. On the contrary, LTM generalizes the idea to allow a mixture of latent topics, which can overcome the word diversity problem of ULM.

LDA shares a same concept as PLSA that uses a set of latent topic variables to model  $P(w|c)$ . The major difference between LDA and PLSA is that PLSA assumes the parameters of topic models to be fixed and unknown, while LDA considers the parameters as random variables that follow a Dirichlet distribution. Because LDA uses a complex form for latent topic modeling, the estimation of model parameters is hard to be solved by an exact inference. To simplify the estimation, a variety of approximation algorithms, such as the variational Bayesian expectation maximization (VBEM) (Blei, 2011; Blei *et al.*, 2003) and Gibbs sampling (Griffiths and Steyvers, 2004) algorithms, have been proposed.

## 2.4 Log-Bilinear Document Modeling

Log-Bilinear document modeling (LBDM) (Maas and Ng, 2010; Maas *et al.*, 2011) can be considered as a relaxed version of LTM. LBDM attempts to learn the word representation with a semantic space and use training documents to constrain those semantically similar words to be represented in near vicinity. The major difference of LBDM and LTM is that LBDM aims to directly parameterize the model for capturing word representations, while LTM focuses on estimating a set of latent topics (Maas and Ng, 2010).

For matching the empirical distribution of words in each document, LBDM introduces a document specific variable  $\theta$  and defines the probability of a document as

$$\begin{aligned} P(d) &= \int P(d, \theta) d\theta \\ &= \int P(\theta) \prod_{w \in d} P(w|\theta)^{n(w,d)} d\theta, \end{aligned} \quad (7)$$

where

$$P(w|\theta) = \frac{\exp(\theta^T \phi_w + b_w)}{\sum_{w' \in \mathbf{V}} \exp(\theta^T \phi_{w'} + b_{w'})}, \quad (8)$$

where  $\phi_w$  is a vector representation of word  $w$ , and  $b_w$  denotes a bias for word  $w$ . LBDM assumes that  $\phi_w$  and  $\theta$  are in  $\mathfrak{R}^\beta$ , and the variable  $\theta$  is modeled by a Gaussian prior density. The probability  $P(w|\theta)$  indicates how close the vector representation of word  $w$ ,  $\phi_w$ , is to  $\theta$ .

Assuming that we are dealing with the entire set of document collection,  $\mathbf{D}$ , the word representation matrix  $\mathbf{R} \in \mathfrak{R}^{\beta \times |\mathbf{V}|}$  (the  $i^{\text{th}}$  column vector of  $\mathbf{R}$  denoting the vector representation of the  $i^{\text{th}}$  word in the vocabulary) and word bias  $\mathbf{b} \in \mathfrak{R}^{|\mathbf{V}|}$  (the  $i^{\text{th}}$  component of  $\mathbf{b}$  denotes the bias term for the  $i^{\text{th}}$  word in the vocabulary) can be estimated by

$$\begin{aligned} \{\hat{\mathbf{R}}, \hat{\mathbf{b}}\} &= \arg \max_{\mathbf{R}, \mathbf{b}} P(\mathbf{D}; \mathbf{R}, \mathbf{b}) \\ &= \arg \max_{\mathbf{R}, \mathbf{b}} \prod_{d \in \mathbf{D}} \int P(\theta_d) \prod_{w \in d} P(w|\theta_d; \mathbf{R}, \mathbf{b})^{n(w,d)} d\theta_d. \end{aligned} \quad (9)$$

The integral over  $\theta_d$  in Eq. (9) is intractable. To simplify the estimation, we assume that the posterior distribution is highly peaked around the MAP estimate of  $\theta_d$ . By adding regularization terms for  $\mathbf{R}$  and  $\theta_d$  and taking the logarithm, the parameters of LBDM are approximately estimated by

$$\begin{aligned} \{\hat{\mathbf{R}}, \hat{\mathbf{b}}\} &= \arg \max_{\mathbf{R}, \mathbf{b}} \left[ \sum_{d \in \mathbf{D}} \sum_{w \in d} n(w,d) \log P(w|\hat{\theta}_d; \mathbf{R}, \mathbf{b}) \right. \\ &\quad \left. + \lambda \|\mathbf{R}\|^2 + \lambda \|\hat{\theta}_d\|^2 \right], \end{aligned} \quad (10)$$

where  $\hat{\theta}_d$  denotes the MAP estimate of  $\theta_d$  for each document  $d \in \mathbf{D}$ . Since the objective function in Eq. (10) is not convex, a coordinate ascent process is performed to estimate the parameters. The estimation process first optimizes the word representations ( $\mathbf{R}$  and  $\mathbf{b}$ ) with keeping the MAP estimates  $\hat{\theta}_d$  of each document fixed. Next, we find the new MAP estimate for each document with keeping the word representations fixed. The two estimation processes are performed iteratively until reach convergence.

When performing DC, for the class  $c$ , the MAP estimate of the variable  $\hat{\theta}_c$  is estimated by using the word representations  $\mathbf{R}$ ,  $\mathbf{b}$ , and a pseudo-document, which is a collection of the entire set of training documents belonging to class  $c$ . Next, the similarity between a document and a class  $c$  is determined by

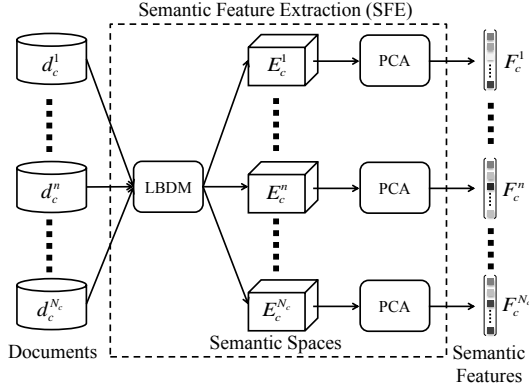


Fig. 1. Semantic feature extraction (SFE) for the  $c^{\text{th}}$  class

$$\begin{aligned}
 P(d|c) &= \prod_{w \in d} P(w|\hat{\theta}_c)^{n(w,d)} \\
 &= \prod_{w \in d} \left( \frac{\exp(\hat{\theta}_c^T \phi_w + b_w)}{\sum_{w' \in \mathbf{V}} \exp(\hat{\theta}_c^T \phi_{w'} + b_{w'})} \right)^{n(w,d)}.
 \end{aligned} \tag{11}$$

### 3 Semantic Naïve Bayes Classifier

NBC with LTM models has taken the semantic information into account and been confirmed effective for DC. However, the “document-level” semantic cues are not directly incorporated. In other words, documents are only treated as a sequence of words when determined the similarity between a class and a document (*cf.* Section 2). To exploit the semantic information of documents, we propose a semantic NBC (SNBC) framework in this paper. In what follows, we articulate the derivation of SNBC and the implementation process of SNBC to perform DC.

#### 3.1 Literal and Semantic Models of SNBC

As presented in Section 2, NBC performs DC by conducting literal term matching (Eqs. (1)-(3)). To incorporate document-level semantic information, we divide  $P(d|c)$  into two parts, namely the literal part and the semantic part, and reformulate Eq. (1) as

$$\begin{aligned}
 \hat{c} &= \arg \max_c P(c|d) \\
 &= \arg \max_c P(d|c)P(c) \\
 &= \arg \max_c P(d_s, d_l|c)P(c),
 \end{aligned} \tag{12}$$

where  $d_s$  and  $d_l$  denote the semantic and literal parts, respectively. By assuming that the literal and semantic parts are conditionally independent given a class, we have

$$P(d_s, d_l|c) = P(d_s|c)P(d_l|c). \tag{13}$$

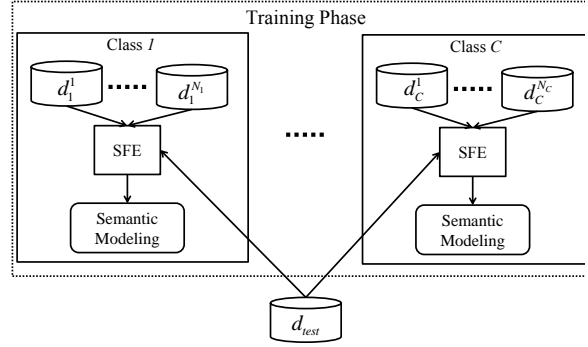


Fig. 2. Semantic modeling and semantic score calculation for a test document

As presented in Section 2,  $P(d_l|c)$  can be estimated by NBC with any language modeling approaches. The calculation of semantic model,  $P(d_s|c)$ , will be detailed in the next section.

#### 3.2 Document-level Semantic Information Capturing

This section describes the semantic feature extraction, semantic modeling, and semantic score calculation procedures in the SNBC framework.

##### 3.2.1 Semantic Feature Extraction

As introduced in Section 2, LBDM can transform a word into a semantic vector representation. In the proposed SNBC framework, we incorporate LBDM to perform semantic feature extraction (SFE). Fig. 1 illustrates the SFE process. Assume that we have  $N_c$  documents in the  $c^{\text{th}}$  class. For the  $n^{\text{th}}$  document, we apply LBDM to represent each word into a semantic vector. The collection of word vectors in that document then forms a semantic subspace for that document, denoted as  $E_c^n$ . Next, we apply principal component analysis (PCA) on  $E_c^n$  to extract its principal vectors,  $F_c^n$ . Finally, we use the principal vectors  $F_c^n$  to capture main directions of semantic topics of the document  $d_c^n$ . In this paper, we only use the eigenvector with the largest eigenvalue as the semantic feature for  $d_c^n$ . We will study the use of multiple eigenvectors in our future work.

##### 3.2.2 Semantic Modeling and Score Calculation

Figure 2 illustrates the semantic modeling process, which can be divided into training and testing stages. In the training stage, we use the semantic features of the training documents to estimate a semantic model, where each class is modeled by a semantic distribution. In this paper, we use the Gaussian mixture model (GMM) for semantic modeling. Because each class may include several sub-topics (e.g., tennis, basketball, and boxing are all categorized in the sports class), we believe that GMM is a suitable model to characterize the semantic distribution for

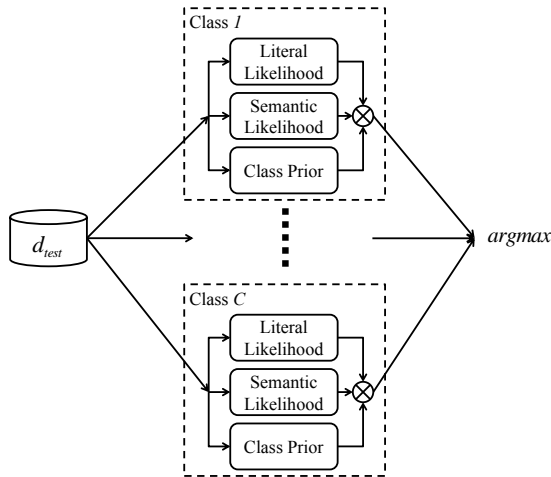


Fig. 3. SNBC score calculation for a test document

each class. The semantic features of the training documents belonging to each class are used to train the GMM for that class. In the testing stage, for each class, the class specific SFE is performed to extract the semantic feature of the test document. Next, the extracted feature is tested on the class specific GMM to obtain the semantic score,  $P(d_s | c)$ , indicating the semantic likelihood of document  $d$  on class  $c$ . Since the covariances for different GMMs may vary, we design a normalization algorithm to compensate the variations of semantic likelihoods by

$$P(d_s | c) = \frac{\sum_{m=1}^{M_c} \pi_m N(F_c | \mu_m, \Sigma_m)}{\sum_{m=1}^{M_c} \pi_m N(\mu_m | \mu_m, \Sigma_m)}, \quad (13)$$

where  $F_c$  is the semantic feature of the test document  $d_{test}$  for the  $c^{\text{th}}$  class, and the GMM model of class  $c$  has  $M_c$  Gaussian components with mean vectors  $\{\mu_1, \dots, \mu_{M_c}\}$ , covariance matrices  $\{\Sigma_1, \dots, \Sigma_{M_c}\}$ , and mixture weights  $\{\pi_1, \dots, \pi_{M_c}\}$ .

### 3.3 SNBC Score Calculation

With NBC and the semantic model, we can calculate the final SNBC score by multiplying scores from three different information sources, namely the class prior, semantic information, and literal language modeling, as illustrated in Fig. 3. We further use  $\alpha$  to control the scale of semantic information. Therefore, the classification rule for SNBC becomes

$$\hat{c} = \underset{c}{\operatorname{argmax}} P(d_s | c)^\alpha P(c) \prod_{w \in d} P(w | c)^{n(w,d)}, \quad (14)$$

where the class prior,  $P(c)$ , is simply kept uniform in this paper.

## 4 Experiments

This section describes our experimental setup, performance measure, and experimental results.

### 4.1 Experimental Setup

We conducted the DC experiments on 20 Newsgroups (20Ng) and WebKB datasets (<http://web.ist.utl.pt/acardoso/datasets/>) (Cardoso-cachopo and Oliveira, 2003). The pre-processing steps include stemming, removing stop words, and removing numbers and words with occurrence below four. 20Ng contains roughly 20,000 documents, which distribute approximately even across 20 classes. These documents are randomly divided into 60% for training and 40% for testing. WebKB originally contains seven different categories, and we use four major classes in the experiments. Finally there are around 4,200 documents, which are randomly divided into two thirds for training and one third for testing. Albeit that the way to systemically determine the values of the parameters in various machine learning approaches is still an open issue and needs further investigation and proper experimentation, the parameters in the following experiments are set empirically as follows. The number of latent topics and the dimension of LDBM are set to 10, which gives the optimal result in our preliminary experiments. For semantic modeling, the number of GMMs equals to the number of classes, and each GMM is characterized by 20 Gaussian components. The parameter  $\alpha$  is set to 0.6.

### 4.2 Performance Measure

In the following DC experiments, we use the standard F1-score measure for evaluation. F1-score ( $F$ ) can be decomposed into two parts, namely recall ( $R$ ) and precision ( $P$ ).

$$R = \frac{\# \text{correct positive prediction}}{\# \text{positive examples}}, \quad (15)$$

$$P = \frac{\# \text{correct positive prediction}}{\# \text{positive prediction}}, \quad (16)$$

$$F = \frac{2 \times R \times P}{R + P}. \quad (17)$$

To evaluate the average F1-scores across all the classes, we adopt micro-averaged and macro-averaged F1-scores (Yang, 1999). The micro-averaged F1-score assigns a same weight across different classes while the macro-averaged F1-score gives each class a specific weight according to the number of documents within that class.

### 4.3 Experimental Results

First, we evaluate the performance of conventional NBC with various language models, including ULM with Jelinek-Mercer smoothing (JM), LDA, and LBDM; their average F1-scores evaluated on WebKB and 20Ng are listed as JM, LDA, and LBDM, respectively, in the upper three rows in Tables 1 and 2. From Tables 1 and 2, we observe that LDA outperforms JM and LBDM in most cases. The results indicate that topic modeling performs better than other language modeling approaches, which is consistent with many previous studies (Blei *et al.*, 2003).

Next, we combine the above three NBC systems, namely JM, LDA, and LBDM, with the semantic information (as presented in Section 3.3); the corresponding results are listed as SNBC (JM), SNBC (LDA), and SNBC (LBDM), respectively, in the lower three rows of Tables 1 and 2. From the experimental results, we note that SNBC consistently outperforms conventional NBC systems. The improvements from NBC to SNBC have been confirmed statistically significant based on the t-test (Agresti and Franklin, 2008). The superscript \* in Tables 1 and 2 indicates that the corresponding improvement is significant at the 0.05 confidence level. Our experimental results confirm that the document-level semantic information provides complementary knowledge to the NBC with LTM and thus improve its performance for DC.

## 5 Conclusions

This paper has proposed a semantic naïve Bayes classifier (SNBC) that incorporates document-level semantic information to improve the performance of the conventional NBC for the DC task. The SNBC framework includes a semantic feature extraction scheme to extract the semantic information of a training/test document and a semantic modeling algorithm to compute the semantic score for a given document. In the testing phase, SNBC combines the semantic score and the language modeling score to perform DC. Our experiments have been conducted on the 20 Newsgroups and WebKB datasets. The results demonstrated that SNBC can improve the DC performance in terms of Micro-F1 and Macro-F1 scores for NBC with various language modeling techniques. The performance improvement of SNBC over NBC confirms the effectiveness of integrating document-level semantic information into the conventional NBC. Notably, this study adopts LBDM to prepare semantic features; other semantic extraction methods, such as PLSA and LDA, can also be used to prepare semantic features. More experiments on SNBC using different semantic extraction methods will be conducted and compared with other existing state-of-the-art approaches in the future.

Table 1. F1-scores of NBC and SNBC on the WebKB dataset

Models	Micro-F1	Macro-F1
JM	0.8381	0.8258
LDA	0.8388	0.8253
LBDM	0.8187	0.8133
SNBC(JM)	0.8491	0.8403
SNBC(LDA)	0.8603	0.8496
*SNBC(LBDM)	0.8488	0.8346

Table 2. F1-scores of NBC and SNBC on the 20Ng dataset

Models	Micro-F1	Macro-F1
JM	0.8131	0.8070
LDA	0.8190	0.8122
LBDM	0.8144	0.8072
SNBC(JM)	0.8293	0.8152
SNBC(LDA)	0.8305	0.8213
*SNBC(LBDM)	0.8346	0.8251

## References

- Alan Agresti and Christine A. Franklin. 2008. In *Statistics: The Art and Science of Learning from Data*. Prentice Hall.
- Alexander Genkin, David D. Lewis, and David Madigan. 2005. Sparse Logistic Regression for Text Categorization. DIMACS Working Group on Monitoring Message Streams. Project Report.
- Ana Cardoso-cachopo, and Arlindo L. Oliveira. 2003. An empirical comparison of text categorization methods. In *String Processing and Information Retrieval*, 10<sup>th</sup> International Symposium, pages 183-196.
- Andrew L. Maas and Andrew Y. Ng. 2010. A probabilistic model for semantic word vectors. In *Deep Learning and Unsupervised Feature Learning Workshop X NIPS*.
- 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 49<sup>th</sup> Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142-150.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993-1022.
- David M. Blei. 2011. Introduction to probabilistic topic models. *Communications of the ACM*, 55(4):77-84.
- Francesco De Comite, Remi Gilleron, and Marc Tommasi. 2003. Learning multi-label alternating decision trees from texts and data. In *Proceedings of MLDM*, pages 251-274.

- Frederic Morin, and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In Proceedings of AISTATS, pages 246-252.
- Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In Proceedings of the Workshop on Pattern Recognition in Practice, pages 381-397.
- Fuchun Peng and Dale Schuurmans. 2003. Combining naïve bayes and n-Gram language models for text classification. Lecture Notes in Computer Science, 2633:335-350.
- Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In the 21st International ACM SIGIR conference on research and development in Information Retrieval, pages 275-281.
- Jerome R. Bellegarda. 2005. Latent semantic mapping. Signal Processing Magazine, IEEE, 22:70-80.
- Jing Bai and Jian-Yun Nie. 2004. Using language models for text classification. In Proceedings of AIRS.
- Meng-Sung Wu and Hsin-Ming Wang. 2012. A term association translation model for naïve bayes text classification. In Proceedings of the 16<sup>th</sup> Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part I, pages 243-253. Springer-Verlag.
- Oh-Woog Kwon and Jong-Hyeok Lee. 2003. Text categorization based on k-nearest neighbor approach for web site classification. Inf. Process. Manage., pages 25-44.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41:391-407.
- Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. In Proceedings of the National Academy of Sciences, 101:5228-5235.
- Thomas Hofmann. 1999a. Probabilistic latent semantic analysis. In Proceedings of Uncertainty in Artificial Intelligence, UAI 99, pages 289-296.
- Thomas Hofmann. 1999b. Probabilistic latent semantic indexing. In Proceedings of the 22<sup>nd</sup> annual international ACM SIGIR conference on Research and development in information retrieval, pages 50-57.
- Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In Proceedings of the 10<sup>th</sup> European Conference on Machine Learning, pages 137-142.
- Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1:67-88.
- Yoshua Bengio, Holger Schwenk, Jean-Sebastien Senecal, Frederic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In Innovations in Machine Learning. Springer Berlin/Heidelberg.



# Cluster-based Web Summarization

Yves Petinot and Kathleen McKeown and Kapil Thadani

Department of Computer Science

Columbia University

503 Computer Science Building

1214 Amsterdam Avenue

New York, New York, 10027

{ypetinot|kathy|kapil}@cs.columbia.edu

## Abstract

We propose a novel approach to abstractive Web summarization based on the observation that summaries for similar URLs tend to be similar in both content and structure. We leverage existing URL clusters and construct per-cluster word graphs that combine known summaries while abstracting out URL-specific attributes. The resulting topology, conditioned on URL features, allows us to cast the summarization problem as a structured learning task using a lowest cost path search as the decoding step. Early experimental results on a large number of URL clusters show that this approach is able to outperform previously proposed Web summarizers.

## 1 Introduction

Abstract Web summaries, which describe the topics and functionalities of Web pages at an *abstract* level, play an essential part in the discovery of new sites and services on the Web. Such summaries are intrinsically difficult to obtain using the content of Web pages. As such, the most successful methods for generating them are effectively extractive and based on the identification of likely abstractive content from linking pages. These *URL-centric* techniques however require a significant amount of redundancy in linking content (Delort et al., 2003).

In this paper, we propose a *summary-centric* approach to Web summarization based on the observation that summaries for similar URLs exhibit both similar content and structure. This similarity is apparent when analyzing summaries from a single ODP category, examples of which are

shown in Table 1. We can see there that summaries of semantically related URLs tend to share common concepts and differ mostly at the level of target-specific attributes. Given a previously unseen URL  $U$ , such a cluster could thus be used as a source of potentially relevant terms for that URL’s summary. In particular, these relevant terms include abstract terms, which may not otherwise be observed in the input data. We propose a graph-based summarization framework that can leverage this phenomenon.

## 2 Proposed Framework

Given a reference cluster  $C$ , we represent the space for summary generation as a graph  $G_C = (V_C, E_C)$ . This graph is obtained by fusing training summaries in  $C$  into a word graph. Each summary  $g_i$  is mapped to a path between the shared source and sink nodes. Each word  $g_i^j$  is thus mapped to a node  $N_k$  and each pair of neighboring words  $(g_i^j, g_i^{j+1})$  to a directed edge  $(N_k, N_l)$ . Notably, we add nodes as needed to guarantee that individual summaries are cycle-free. Figure 1 shows a simple summary graph.

### 2.1 Node Alignment

During the graph construction, nodes from distinct paths are iteratively combined to elicit the structural and content commonalities between summaries in the reference cluster. Following (Filippova, 2010) unambiguous nodes — i.e. nodes whose surface form match exactly and for which only one candidate exists — are always aligned, while ambiguous nodes are aligned to the candidate node with maximum context overlap. When there is no context overlap, a new node is added to the graph.

URL	Abstract Summary
<a href="http://www.qgazette.com/">http://www.qgazette.com/</a>	Published weekly for the Queens, New York community. Includes information on politics, religion, dining, seniors, events, archives, classifieds and subscription details.
<a href="http://www.queenschronicle.com/">http://www.queenschronicle.com/</a> <a href="http://www.rockawave.com/">http://www.rockawave.com/</a>	Local Queens NY news classifieds. Published weekly in Rockaway, featuring local news, sports, community calendar, classified ad section, archives and subscription and advertising details.
<a href="http://www.observer.com/">http://www.observer.com/</a>	Online version of the newspaper, providing coverage of local politics and media, Real Estate, fashion, and the Arts.
<a href="http://www.nytimes.com/">http://www.nytimes.com/</a>	Online edition of the newspaper's news and commentary. [Registration required]

Table 1: Sample entries from the ODP category /News/Newspapers/Regional/United\_States/New\_York. All entries in this category describe sites of news organization located in the New-York area.

## 2.2 Template Slots

The content of reference summaries is likely to be only partially relevant to a previously unseen URL. In particular, certain paths in the summary graph may contain nodes whose surface form is target-specific. We use the following heuristic to decide on the presence of template slots in a summary:

- Adverbs, adjectives and named entities are treated as slots.
- Any term occurring in at least 25% of paths will not be treated as a slot;

Similarly to (Barzilay and Lee, 2003), slot identification is performed prior to alignment.

## 3 Features

In this section we present the feature sets used to condition the summary graph topology on a target URL  $U$ . Two aspects of the graph need to be trained, namely edge costs and slot locations. We introduce features for both.

### 3.1 Edge Feature Templates

We use the following feature templates to represent the compatibility between  $U$  — represented by its text modalities, as listed in Table 2 — and a specific edge in the summary graph.

**Edge prior** Probability of appearance of edge  $e_{ij}$  in reference paths (summaries).

**Edge appearance + Modality** Frequency of edge  $e_{ij}$  in each modality  $M_k$ . We consider that  $e_{ij}$  appears in  $M_k$  if its source and sink co-occur in  $M_k$ .

**Source/Sink prior** Probability of Source/Sink node  $N_i$  in reference paths (summaries).

**Source/Sink appearance + Modality** Indicates whether Source/Sink node  $N_i$  occurs in  $M_k$ .

**Modality + n-gram** Compatibility between the edge  $e_{ij}$ , the modality  $M_k$  and the n-gram  $n_l$ , where  $n$  is in the range  $[1, 3]$ .

### 3.2 Slot Features

To allow for the use of supervised methods in learning optimal edge costs we need a graph whose structure remains unchanged from one training instance to the next. The fillers of slot locations are thus described using features that are not surface-based:

**Semantic Type** We only consider fillers compatible with the host slot.

**Modality appearance** Frequency of filler candidate in modality  $M_k$ .

**Content HTML Context** HTML (Style + Structural) context of filler candidate in the content of  $U$ .

## 4 Learning Model

Using the summary graph topology and the features defined above we express the abstract Web summarization task as a structured learning problem. Specifically, we seek to obtain edge weights such that the cost of individual reference summaries — which, as we saw earlier, are mapped to paths — is minimized. Given a set of features  $\mathcal{F}$ , the optimal set of feature weights  $w^*$  is such that:

$$w^* = \arg \min_w \sum_{g \in \mathcal{R}(U)} Cost_w(g) \quad (1)$$

Since our core constraint in building the summary graph is that each summary should map to a cycle-free path, identifying the optimal summary given a set of edge weights reduces to solving a lowest cost path problem:

$$w^* = \arg \min_w \sum_{g \in \mathcal{R}(U)} \sum_{i=1}^{|g|} \sum_{f_k \in \mathcal{F}} Cost_{f_k}(e_{g_{i-1}g_i}) \quad (2)$$

Modality	Feature Types	Description
URL content	$n$ -gram ( $n \in [1, 3]$ ) $n$ -gram + context	$n$ -grams generated from the target page content $n$ -gram with HTML context (immediately surrounding HTML tag)
URL title	1-gram	1-grams generated from the target URL title
URL words	1-gram	1-grams generated from the target URL string (e.g. "nytimes", "com")
URL anchor text	$n$ -gram ( $n \in [1, 3]$ )	$n$ -grams generated from the anchor text for the target URL

Table 2: Modality and features used to represent a target URL  $U$ .

In this setting, generation is achieved by running the decoder using the optimal set of edge weights.

#### 4.1 Structured Perceptron

One way to solve this learning problem is to use a structured perceptron algorithm (Collins, 2002) where weights  $w$  control the linear contribution of individual features to the aggregate cost of edges.

$$Cost_{f_k}(e_{ij}) = w_k^{e_{ij}} \cdot f_k \quad (3)$$

The structured perceptron algorithm is shown in Algorithm 1. At every iteration, decoding is achieved via a search for the shortest path based on the edge costs induced by the current weights  $w^*$ . Following recent work on structured learning (Huang et al., 2012), we do not require this search to be exact, but to guarantee that each iteration results in a valid update. Our decoder thus uses beam search (beam size  $b = 5$ ) combined with an early update procedure, the latter helping in significantly speeding up model training.

---

#### Algorithm 1 Structured Perceptron

---

**Require:**  $\{u^i, g^i\}_{i=1}^n$   
1:  $w^* \leftarrow \{0\}$   
2: **for**  $i = 1 \rightarrow T$  **do**  
3:   **for**  $j = 1 \rightarrow n$  **do**  
4:      $g^* \leftarrow ShortestPath_{G^{w^*}}(u^i)$   
5:      $w^* \leftarrow w^* + \phi(u^i, g^i) - \phi(u^i, g^*)$   
6:   **end for**  
7: **end for**

---

#### 4.2 Slot Filling

During the inference phase, we substitute slot locations with alternate paths, each containing a candidate filler for the slot. Each of these paths is associated with the slot features discussed earlier, however the feature weights of each slot are shared, thus allowing the learning algorithm to converge towards appropriate weights for filler selection.

### 5 Evaluation

We compare the performance of our model against a reimplementations of the two summarization al-

gorithms - content-based and context-based - proposed in Delort et al. (2003). We apply our summarization algorithms and the baselines to a random sample of 56 ODP categories comprising at least 50 entries. For each category, we split the set of available summaries into training (90%) and testing (10%), train the summarization algorithms on the training set, and report (macro) average performance on the testing set.

ROUGE (Lin, 2004) results comparing both our proposed summarization model and the baseline systems to the ODP ground truth are provided in Table 3. The summary-graph model, both with and without slot-filling, shows significant improvements compared to the baselines in terms of ROUGE-1 and ROUGE-L scores. ROUGE-2 performance, however, is not on par with the baselines. Long-distance sequence similarity (ROUGE-L) being higher, we believe this could indicate the inability of our model to capture target-specific bi-grams that have little or no support in the training summaries. Allowing the topology of the summary-graph to adapt to its target, for instance by introducing missing edges supported by the input data, should help alleviate this issue. Finally, the performance of the model with slot filling shows slight improvement over the basic model, however we observed that the system frequently fails in extracting slot fillers. Future work will focus on acquiring more filler candidates and better features to model them.

### 6 Previous and Related Work

The work presented in this paper is linked to previous research in Web summarization and T2T language generation. Most works on the former have been on extractive methods, owing to the complexity of Web content but also to the need for compressed versions of Web pages. Other works have in fact eluded the question of generation to instead focus on the extraction of salient keywords from Web sites (Glover et al., 2002; Zhang et al., 2004). In the context of Web search engines, the compression task is constrained further by the amount of *SERP real estate* available for any single snip-

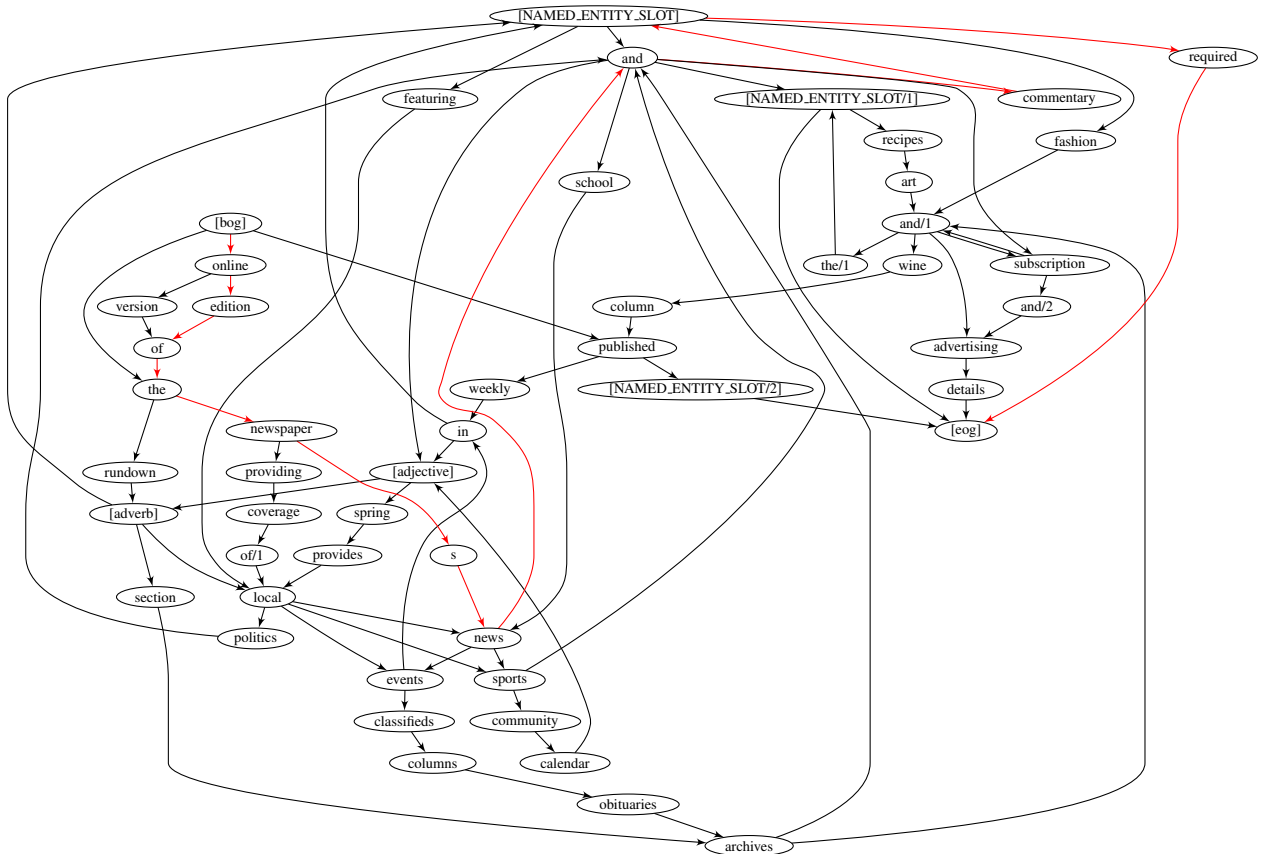


Figure 1: Summary graph associated with a subset of /News/Newspapers/Regional/United\_States/New\_York. The path for the URL <http://www.nytimes.com> is shown in red

Summarization System	ROUGE-1	ROUGE-2	ROUGE-L
Delort - Content	0.07163	0.04492	0.06574
Delort - Context	0.06783	0.03979	0.06197
Summary-graph	0.16222†	0.02775	0.14370†
Summary-graph + slot filling	0.16832†	0.02702	0.14729†

Table 3: Performance of summarization algorithms. † indicates statistically significant improvements (according to a paired t-test with  $p < 0.05$ ) compared to the provided baselines.

pet and how content may be truncated (Clarke et al., 2007). Sun et al. (2005), in particular, leveraged the ODP hierarchy to mitigate data scarcity for certain URLs, however they did not exploit ODP summaries themselves. Several efforts have focused on producing Web summaries using the content of linking pages as a source of descriptive content. Amitay and Paris (2000) assumes full summaries can be readily found on a single page linking to the target site. Delort et al. (2003) makes less stringent assumptions and seeks to combine descriptive content from multiple linking pages. Closer to our work, Berger and Mittal (2000) proposed a generative solution embracing the noisiness of Web data and trained directly over ODP (URL,summary) pairs. Finally our work relates to T2T generation as we seek to generate well-

formed sentences without resorting to semantic representations of either the input or output contents. Graph-based models similar to ours have been used for tasks ranging from string reconstruction (Wan et al., 2009) to sentence fusion and compression (Filippova and Strube, 2008; Filippova, 2010).

## 7 Conclusion and Future Work

We have introduced a word graph model for the task of Web summarization, and showed that per-cluster word graphs make it possible to combine abstractive and extractive behaviors. A limitation of our model is the need for existing reference clusters from which we build our summary graphs. Future work will investigate the dynamic production of such clusters.

## References

- Einat Amitay and Cecile Paris. 2000. Automatically summarising web sites - is there a way around it? In *CIKM 2000*, pages 173–179.
- Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT, 2003*, pages 16–23.
- A. Berger and V. Mittal. 2000. Ocelot: a system for summarizing web pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 144–151.
- Charles L.A. Clarke, Eugene Agichtein, Susan Dumais, and Ryan W. White. 2007. The influence of caption features on clickthrough patterns in web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 135–142.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, page ...
- Jean-Yves Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. 2003. Enhanced web document summarization using hyperlinks. In *Hypertext 2003*, pages 208–215.
- K. Filippova and M. Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, Hawaii, Association for Computational Linguistics*, page 177–185.
- Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of COLING 2010*, pages 322–330.
- Eric J. Glover, Kostas Tsioutsoulouklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. 2002. Using web structure for classifying and describing web pages. In *Proceedings of WWW 2002*, pages 562–569.
- Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter for the Association for Computational Linguistics: Human Language Technologies*, pages 142–151.
- Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26*.
- Jian-Tao Sun, Dou Shen, Hua-Jun Zeng, Qiang Yang, Yuchang Lu, and Zheng Chen. 2005. Web-page summarization using clickthrough data. In *SIGIR 2005*, pages 194–201.
- Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2009. Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 852–860.
- Y. Zhang, N. Zincir-Heywood, and E. Milios. 2004. World wide web site summarization. In *Web Intelligence and Agent Systems, 2(1)*, pages 39–53.

# Automated Activity Recognition in Clinical Documents

**C. Thorne M. Montali D. Calvanese**  
Free University of Bozen-Bolzano  
Bolzano, Italy  
surname@inf.unibz.it

**E. Cardillo C. Eccher**  
Fondazione Bruno Kessler  
Trento, Italy  
surname@fbk.edu

## Abstract

We describe a first experiment on the identification and extraction of computer-interpretable guideline (CIG) components (activities, actors and consumed artifacts) from clinical documents, based on clinical entity recognition techniques. We rely on MetaMap and the UMLS Metathesaurus to provide lexical information, and study the impact of clinical document syntax and semantics on activity recognition.

**Keywords.** Clinical entity recognition, computer interpretable guideline, UMLS Metathesaurus.

**Introduction.** Clinical practice guidelines are systematically developed documents specifying the activities, resources and personnel required to cure or treat a specific illness or medical condition (Field and Lohr (1990)). The need to instantiate them into clinical protocols and workflows has given rise to *computer-interpretable guidelines* (CIGs) (De Clercq et al. (2008)), i.e., formal representations of the care process or plan, and to several natural language processing (NLP) techniques aimed at automating the costly manual CIG generation process (Kaiser et al. (2007), Serban et al. (2007)). All NLP approaches leverage on annotated biomedical resources (e.g., the CLEF corpus from Roberts et al. (2007) and Mykowiecka and Marciniak (2011)), or on frameworks such as cTAKES (Savova et al. (2010)). The key lexical-semantic resource in this domain is the US National Library of Medicine’s Unified Medical Language System (UMLS) Metathesaurus (Bodenreider (2004)), complemented by its front-end MetaMap (Aronson and Lang (2010)).

In this paper we conduct a first experiment on how to apply entity recognition techniques inspired by Abacha and Zweigenbaum (2011), to

recognize CIG components in medical documents. The process dimension of CIGs consists of four pillars: **(1)** *activities* to be executed; **(2)** the *resources* they use or consume; **(3)** the *actors* that execute them; **(4)** *control flows and gates* that temporally constrain activities. We focus in this paper on activities, the main building block of CIGs, and to a lesser extent on resources and actors. All these components are denoted by content words and can be used to build CIG fragments. We rely on MetaMap annotations and evaluate our techniques over an UMLS-annotated clinical corpus.

**CIGs and Activities.** Activities are entities difficult to identify with current resources: within clinical documents, in fact, not only verbs (**VBs**) but also proper nouns (**PNs**), common nouns (**NNs**) and, more in general, noun phrases (**NPs**)<sup>1</sup> can refer to them. Figure 1 shows an example from the type-2 diabetes guideline of the National Institute for Health and Clinical Excellence (NICE) (NICE - NHS (2009)) expressing a conditional CIG/process fragment, annotated automatically with MetaMap. To correctly extract the “deep” intended representations it is necessary to recognize that the two entities “blood glucose control” and “oral glucose-lowering medication” are activity tokens. MetaMap annotations provide a clue, but we still need to “filter out” the “clinical attribute” UMLS annotation. We want to understand how this information can be used for this task within an entity recognition framework.

**Clinical Entity Recognition.** Let  $\vec{c}$  denote a vector of clinical *entity type labels*, and  $\vec{\alpha}$  a vector of input *noun phrases (NPs)* or *entities*. The goal of *clinical entity recognition*, see Abacha and Zweigenbaum (2011), can be formulated as the task of finding the best scoring vector of clinical

<sup>1</sup>In this paper we refer to the Penn Treebank part-of-speech (POS) notation as described by Marcus et al. (1993).

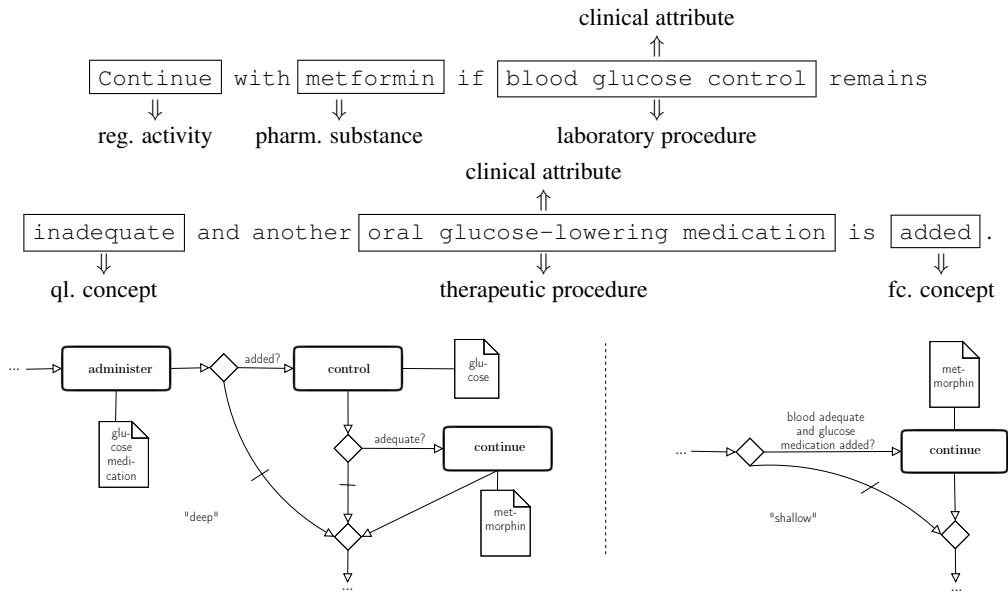


Figure 1: **Top:** MetaMap UMLS (automated) annotations of the NICE diabetes guideline fragment; boxes surround entities, annotations are MetaMap’s. **Bottom:** Two candidate CIG fragments (represented in Business Process Modeling Notation (BPMN), see Ko et al. (2009)): to the left, the intended “deep” CIG, to the right a “shallow” CIG. Control flows (diamonds) specify the acceptable orderings of the activities (rounded rectangles); activities consume resources (folded-corner rectangles).

entity type labels:  $\vec{c}^* = \arg \max \{ \vec{c} \mid \mu(\rho(\vec{\alpha}, \vec{c})) \}$ , where  $\mu(\cdot)$  denotes a *recognizer* built using a classification model (e.g., a logistic regression algorithm), and  $\rho(\cdot, \cdot)$  is a feature extraction function. In the following paragraphs we study this task w.r.t. the set {activity, resource, actor, other} of entity types.

**The SemRep corpus.** Since no UMLS annotated clinical guideline corpora are available for research purposes, we ran our experiments over the SemRep corpus by Kilicoglu et al. (2011), a small annotated clinical corpus whose domain largely overlaps with that of guidelines. It consists of 500 clinical excerpts (MedLine/PubMed) and contains 13,948 word tokens manually annotated by clinicians and domain experts, covering the whole clinical domain. UMLS concept types annotate a total of 827 NPs.

**Features.** The focus of our experiments is to understand the predictive power of syntax and semantics for CIG entity recognition, and in particular for activity recognition. Intuitively, both syntax and semantics can contribute to the prediction of clinical entity types, but it is not a priori clear which one contributes more. Similarly to Zhou and He (2011) we used the Stanford parser (see Klein and Manning (2003)) to extract syn-

tactic features, and MetaMap to extract semantic features. We harvested clinical types by mapping UMLS concept types returned by MetaMap to their subsuming clinical types. In the top of Table 1 we show a sample of UMLS concept types subsumed by “activity”, “resource”, “actor” and “other”, whereas in its bottom we summarize the extracted features, described in detail below.

By mining the NPs sentence parse trees, we extracted the following syntactic features: depth of nesting (*nest*); position in the phrase (*pos*); occurrence in a subordinated phrase (*sub*). The intuition behind these features is that certain types may correlate strongly with syntax (e.g., one would expect “resource” to annotate an object NP).

The semantic features were extracted by computing several measures of label overlap and frequency. The rationale of these features is that, while MetaMap outputs many possible clinical meanings of the constituent NNs of an NP entity, giving rise to multiple “activity”, “resource”, “actor” and “other” annotations per NN and NP, it tends to output meanings that are semantically related (within the UMLS Metathesaurus hierarchy) to the NP’s intended type.

We measured the raw frequency *freq* of the NP entity type *c* in the SemRep corpus, the degree of annotation overlap *hd* between the bag of possi-

activity	actor	resource	other
laboratory procedure	professional society	manufactured object	qualitative concept

feature $F$	description	value $f$
<i>nest</i>	nesting level in tree	integer $\in \mathbb{N}$
<i>pos</i>	position w.r.t. verb	subject, predicate
<i>sub</i>	occurs in clause?	yes, no
<i>freq</i>	freq. of label in corpus	integer $\in \mathbb{N}$
<i>lf</i>	rel. freq. of label in <b>NP</b>	real $\in [0, 1]$
<i>hd</i>	head/ <b>NP</b> overlap	real $\in [0, 1]$
<i>ls</i>	label/ <b>NP</b> overlap	real $\in [0, 1]$
<i>class</i>	<b>NP</b> entity type	act., actor, res., other

Table 1: **Top:** CIG entity labels and sample UMLS concept types they subsume. **Bottom:** **NP** features considered; the class label is the *dependent* feature we want to predict.

bly repeated labels *labs* collected using MetaMap from all the **NNs** in an **NP**, and the bag of possibly repeated labels of its head noun *labsh*. In addition, we computed the relative frequency *lf* of the **NP** entity type  $c$  w.r.t. *labs*:

$$hd = \frac{||labs \cap labsh||}{||labs|| + ||labsh||} \quad lf = \frac{||labs \cap \{c\}||}{||labs||} \quad (1)$$

where  $|| \cdot ||$  and  $\cap$  denote resp. bag cardinality and intersection. The intuition behind these two features is that the intended type will tend to prevail within the annotations of an **NP**, and in particular among its head **NN** and its modifiers. Finally, we took into account the taxonomical structure of the UMLS Metathesaurus and defined the following label/**NP** overlap *ls*:

$$ls = \frac{||labs \cap sub(c)||}{||labs|| + ||sub(c)||} \quad (2)$$

where *sub(c)* is the bag of all the UMLS concept types that are subsumed by the entity type label  $c$ . The *ls* feature measures how similar are the MetaMap **NP** annotations to the UMLS hierarchy subsumed by  $c$ . In all cases, a simple Laplace smoothing was applied.

**Evaluation Framework.** In our experiments the main goal was to evaluate activity recognition features rather than classifier design and evaluation. We thus relied on standard classification models from the known Weka<sup>2</sup> data mining framework. We trained and evaluated the following classifiers: (i) logistic classifier (Logit), (ii) support vector machine (SVM), (iii) naive Bayes classifier

<sup>2</sup>[www.cs.waikato.ac.nz/~ml/weka/](http://www.cs.waikato.ac.nz/~ml/weka/)

(Bayes), (iv) neural network (Neural), and (v) decision tree (Tree). To measure the significance of each single feature, we removed each time a feature  $F_i$  from the space  $\{F_1, \dots, F_7\}$  of syntactic and semantic *independent* features from Table 1 and retrained and reevaluated the classifiers w.r.t. the feature space  $\{F_1, \dots, F_{i-1}, F_{i+1}, \dots, F_7\}$ .

In parallel to this, we studied the impact of context over activity recognition, and its interplay with our features. To this end we considered a baseline scenario, in which context is restricted to **NPs**, and a scenario in which we take into consideration all the annotated **NPs** of a SemRep sentence. This distinction is important since SemRep is a small and sparsely annotated corpus, for which enhanced feature spaces may not prove informative. These two scenarios were modeled as follows. **(1)** A set of **NP** observations: for each **NP**  $\alpha$  in SemRep, we extracted the feature vector  $(f_1^\alpha, \dots, f_7^\alpha, c^\alpha)^T$ . **(2)** A set of sentence observations: for each vector  $(\alpha_1, \dots, \alpha_k)^T$  of annotated **NPs** in a SemRep sentence, we extracted feature vectors  $(f_1^{\alpha_1}, \dots, f_7^{\alpha_1}, c^{\alpha_1}, \dots, f_1^{\alpha_k}, \dots, f_7^{\alpha_k}, c^{\alpha_k})^T$ .

For each combination of classifier feature and scenario, we performed a 10-fold cross-validation to measure precision (Pr), recall (Re), F1-measure, and the overall accuracy (Ac) of the classifiers for the activity recognition task<sup>3</sup>.

**Results and Discussion.** The baseline scenario (see Figure 2, left) shows a drop in average precision, recall, F-measure and accuracy when *hd* and *freq* are disregarded, and a minor drop when *ls* is disregarded. The removal of syntactic features on the other hand has a smaller effect. Considering sentence context (see Figure 2, center), we can observe a greater impact for *sub*, and a minor drop when *ls* is disregarded. But sentence context gives rise also to a clear decrease in average classifier performance. Thus *sub*, while significant, is less useful than the semantic features.

This last observation is substantiated by corpus evidence. One way to see how, is to focus on the distribution of syntax relatively to corpus domain. Syntactic structures can be approximated by function words<sup>4</sup> (e.g., subordinators (**INs**) such as “if”

<sup>3</sup>For reasons of space, we present here a summary of the results obtained; for a more detailed description, please refer to [www.inf.unibz.it/~cathorne/vericlig/ijcnlp2013-exp.pdf](http://www.inf.unibz.it/~cathorne/vericlig/ijcnlp2013-exp.pdf)

<sup>4</sup>For the POS tagging we relied on a Natural Language Toolkit (NLTK) 3-gram tagger by Bird et al. (2009), trained over the (POS annotated) Brown corpus.



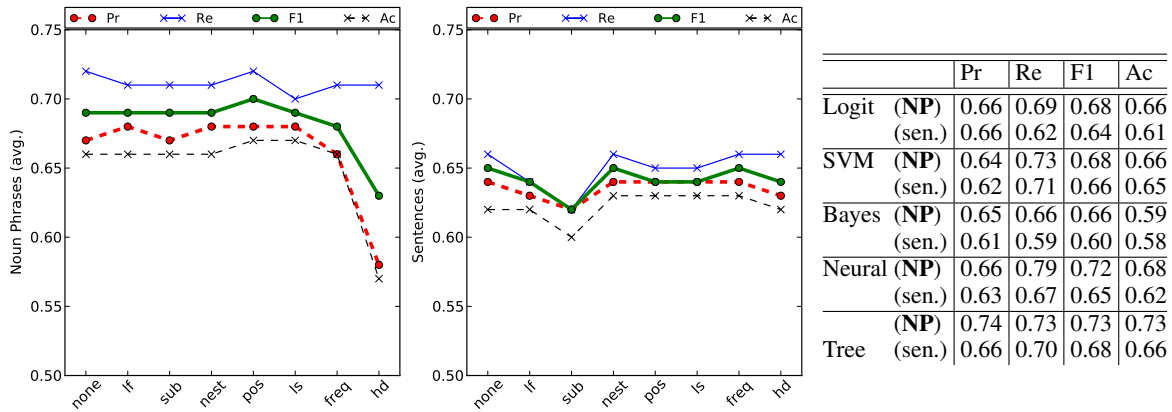


Figure 2: **Left, Center:** Results of 10-fold cross-validation by scenario. On the  $y$ -axis, activity recognition precision, recall, F1-measure and classifier accuracy (classifier averages). On the  $x$ -axis, the feature(s) removed. The tag “none” means that no feature was removed. **Right:** Results for the original (complete) feature space, by classifier and label context (noun phrase **NP** or sentence *sen.*).

corpus	size (words)	domain	rel. freq.
Brown	1,391,708	news	0.16
Friederich	3,824	processes	0.17
SemRep	13,948	clinical	0.18
diabetes2	7,109	clinical	0.16
eating dis.	5,078	clinical	0.17
schizophrenia	5,367	clinical	0.18

$\chi^2$	$p$	df.	$t$ -score	$p$	df.
43.13	0.00	2	1.03	0.36	5

Table 2: **Top:** Function word relative frequency across corpora and domains. **Bottom:** Statistical tests ( $\chi^2$ -test of independence and  $t$ -test).

or “then”, coordinators (**CCs**) such as “or”).

We compared to SemRep: (i) a subset of the Brown corpus (Francis and Kucera (1964)), (ii) a corpus of business process specifications (Friederich et al. (2011)), (iii) a subset of the NICE diabetes-2 guideline (NICE - NHS (2009)), (iv) a subset of the NICE eating disorders guideline (NICE - NHS (2004)), and (v) a subset of the NICE schizophrenia guideline (NICE - NHS (2010)). We run the following statistical tests (see Gries (2010)) at  $p = 0.01$  significance: **(1)** a  $t$ -test (null hypothesis: cross-corpora function word mean relative frequency is 0.20); **(2)** a  $\chi^2$ -test of independence (null hypothesis: function word distribution is correlated to corpus domain). The test results (see Table 2) show that syntax is uniform across domains, and thus has a more limited impact relatively to semantics.

Syntax, however, can be leveraged to optimize prediction results when exploited by classifiers

sensitive to categorical data. The classifier that performed better overall was the decision tree (see Figure 2, right), which seems to exploit better the more limited impact of *sub*, *pos*, and *nest*.

**Conclusions and Further Work.** We have conducted preliminary experiments on automatic clinical activity recognition using MetaMap and entity recognition techniques. We experimented our techniques on the SemRep gold standard UMLS-annotated corpus. Our experiments suggest that the semantic environment of an entity is more useful for this task. Corpus analysis on SemRep and other corpora seems to confirm this observation. In the future, we plan to consider more powerful classification models for NLP, such as conditional random fields (CRFs), able to exploit possible dependencies among features. We plan to focus on document semantics, by considering more complex semantic features (based on, e.g., thesaurus-based similarity metrics). Finally, to better cope with data sparseness we intend to consider a bigger corpus by integrating SemRep with, e.g., the i2b2 clinical corpus as suggested by Abacha and Zweigenbaum (2011).

**Acknowledgments.** The present work has been done within the context of the VERICLIG project<sup>5</sup>, supported by a grant from the Free University of Bozen-Bolzano Foundation.

<sup>5</sup>[www.unibz.it/~cathorne/vericlig](http://www.unibz.it/~cathorne/vericlig)

## References

- Asma Ben Abacha and Pierre Zweigenbaum. 2011. Medical entity recognition: A comparison of semantic and statistical methods. In *Proceedings of the BioNLP 2011 Workshop*.
- Alan R. Aronson and François-Michel Lang. 2010. An overview of MetaMap: Historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O’Reilly.
- Olivier Bodenreider. 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32:D267D270.
- Paul De Clercq, Katharina Kaiser, and Arie Hasman. 2008. Computer interpretable medical guidelines. In A. Ten Teije et al., editor, *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, chapter 2, pages 22–43. IOS Press.
- Marilyn J. Field and Kathleen N. Lohr, editors. 1990. *Clinical Practice Guidelines. Directions for a New Program*. National Academy Press.
- Nelson Francis and Henry Kucera. 1964. A standard corpus of present-day edited american english, for use with digital computers. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, USA.
- Fabian Friederich, Jan Mendling, and Frank Puhlmann. 2011. Process model generation from natural language text. In *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011)*.
- Stefan Th. Gries. 2010. Useful statistics for corpus linguistics. In Aquilino Sánchez and Moisés Almela, editors, *A mosaic of corpus linguistics: selected approaches*, pages 269–291. Peter Lang.
- Katharina Kaiser, Cem Akaya, and Silvia Miksch. 2007. How can information extraction ease formalizing treatment processes in clinical practice guidelines? A method and its evaluation. *Artificial Intelligence in Medicine*, 39(2):151–163.
- Halil Kilicoglu, Graciela Rosenblat, Marcelo Fisman, and Thomas C. Rindfleisch. 2011. Constructing a semantic predication gold standard from the biomedical literature. *BMC Bioinformatics*, 12(486).
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics ACL 2003*.
- Ryan K.L. Ko, Stephen S.G. Lee, and Eng Wah Lee. 2009. Business process management (BPM) standards: A survey. *Business Process Management Journal*, 15(5):744–791.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.
- Agnieszka Mykowiecka and Malgorzata Marciniak. 2011. Some remarks on automatic semantic annotation of a medical corpus. In *Proceedings of the 3rd International Workshop on Health Document Text Mining and Information Systems*.
- NICE - NHS. 2004. Eating disorders. Available from <http://www.nice.org.uk/nicemedia/live/10932/29218/29218.pdf>.
- NICE - NHS. 2009. Type 2 diabetes. Available from <http://www.nice.org.uk/nicemedia/pdf/CG87NICEGuideline.pdf>.
- NICE - NHS. 2010. Schizophrenia. Available from <http://www.nice.org.uk/nicemedia/pdf/CG87NICEGuideline.pdf>.
- Angus Roberts, Robert Gaizaskas, Mark Hepple, Neil Davis, George Demetriou, Yikun Guo, Jay Kola, Ian Roberts, Andrea Setzer, Archana Tapuria, and Bill Wheeladin. 2007. The CLEF corpus: Semantic annotation of a clinical text. In *Proceedings of the AMIA 2007 Annual Symposium*.
- 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.
- Radu Serban, Anette ten Teije, Frank van Harmelen, Mar Marcos, and Cristina Polo-Conde. 2007. Extraction and use of linguistics patterns for modelling medical guidelines. *Artificial Intelligence in Medicine*, 39(2):137–149.
- Deyu Zhou and Yulan He. 2011. Semantic parsing for biomedical event extraction. In *Proceedings of the 9th International Conference on Computational Semantics IWCS 2011*.

# Large-scale text collection for unwritten languages

Florian R. Hanke and Steven Bird

Department of Computing and Information Systems, University of Melbourne  
florian.hanke@gmail.com, sbird@unimelb.edu.au

## Abstract

Existing methods for collecting texts from endangered languages are not creating the quantity of data that is needed for corpus studies and natural language processing tasks. This is because the process of transcribing and translating from audio recordings is too onerous. A more effective method, we argue, is to involve local speakers in the field location, using an audio-only translation interface that is portable and easy to use. We present encouraging early results of an experimental investigation of the efficiency of creating translations using this method, and report on the quality of the resulting content.

## 1 Introduction

Language documentation aims to “provide a comprehensive record of the linguistic practices characteristic of a given speech community” (Himmelman, 1998). In a typical language documentation workflow, a linguistic event is recorded, then metadata is added concerning participants, location, language, and so forth. Later, the recording is transcribed, glossed at the word or morpheme level, and then a translation is provided. Not all of these activities occur in the field: usually recording, metadata capture, and some transcription work take precedence over word-level glosses and phrasal translations (Thieberger, 2011).

Woodbury (2007) argues that for an archive entry to be analysable for a future linguist, multiple kinds of translations are needed, for example audio recordings of UN-style simultaneous oral translations, or sentence by sentence translations. The typical workflow of documentary linguistics does not produce the amount of data required for large-scale corpus-based analysis of the language once it is no longer spoken (Abney and Bird,

2010). In addition, the typical workflow necessitates the creation of transcriptions before any annotations can be made, for example in ELAN,<sup>1</sup> a popular software tool for linguistic annotation (Berez, 2007).

We propose to add a new path to this workflow (see Figure 1) to facilitate crowdsourcing of translations, whether by local (typically village-based) speakers or by geographically distributed speakers from the diaspora (Reiman, 2010; Bird, 2010; Zaidan and Callison-Burch, 2011). To this end, we have developed mobile phone software with easy-to-use interfaces for collecting oral translations.

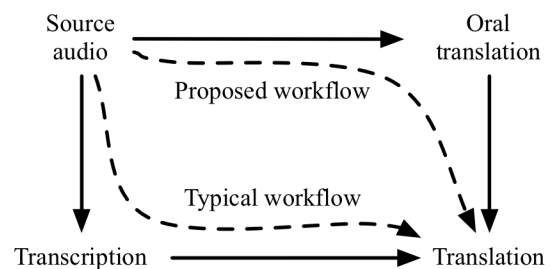


Figure 1: Current and proposed workflows for early oral annotation.

In this paper, we focus on a single activity, oral translation, via one of the mentioned interfaces, a “phonecall” interface. We have developed software that runs on mobile phones, and which has been successfully deployed in off-grid indigenous village settings. It can be used by linguists and native speakers to rapidly collect a substantial amount of high-quality, time-aligned bilingual audio. We report on an experimental investigation of the efficiency of creating translations, and the quality of the translated content.

<sup>1</sup><http://tla.mpi.nl/tools/tla-tools/elan/>

## 2 Oral translation using Aikuma

Oral translation is the process of listening to a segment of audio in a source language and spontaneously producing a spoken translation in a second language. Typically, a body of recordings has already been collected in the (unwritten) source language, and the content of these recordings is to be made accessible to speakers of a more widely spoken language.

Aikuma is an Android application which supports the recording of audio sources, along with phrase-by-phrase oral translation.<sup>2</sup> Aikuma aims to make the recording of high quality translations a simple and natural process. To achieve naturalness, we adopt the metaphor of a phone call. The process does not use the touch screen or any buttons, but relies exclusively on audio and proximity sensors for control.

For example, let us assume we have an original recording of someone telling a story. As soon as this user holds the phone up to his or her ear, the original recording will start playing. This is achieved by using the proximity sensor present in all Android phones. The recording will continue to play as long as the user holds the phone up to their ear. At any moment during the recording, the user is free to speak their translation of what they have heard. The phone is continually monitoring the microphone input and as soon as the user starts speaking, the phone stops playback and begins recording. This enables a wide range of translations: UN-style oral free, phrase-by-phrase, or literal translations (Woodbury, 2007). When the user stops speaking for two seconds, the phone stops recording. It then rewinds the source recording by 650 ms, to ensure that the user does not miss any speech that overlapped with the segment boundary. Finally, it resumes playing the next part of the source. The process is repeated until the end of the source.

The phone's storage now contains the source audio file, along with the translation file and a mapping file. The translation file contains the concatenated recordings of the oral translations. The mapping file specifies how each segment of oral translation corresponds to the source audio. Users can listen to the original, or the translation, or interleaved playback of the original with the translation.

To evaluate our approach, we performed an

<sup>2</sup><http://github.com/langtech/aikuma>

experiment, then made improvements, then performed a second experiment. Both experiments were conducted in March 2013 in the Interface Design Laboratory at the University of Melbourne.

## 3 Experiment 1

### 3.1 Subjects and Materials

The participants of experiment 1 were seven Brazilian university students, aged 19 to 31. All had received four years of instruction in English as a second language.

The following procedure was carried out. Participants were given a one-minute demonstration of the Aikuma app. Then they were free to try it for up to two minutes on a test recording. We used low-end Huawei phones with a touchscreen. As an original source recording, we used an interview of Brazilian Tom Jobim, dating from the 1980s.<sup>3</sup> The participants then used the interface to translate the original source recording.

### 3.2 Results

The efficiency of the system was surprisingly high: on average, a translation of the 6:19 min long original required 6:38 min. Total length ranged from 12:05 min to 14:31 min, with an average of 12:57 min, a factor of 2.07 times the original's length.

This was a far lower duration than we expected, as the provided translations included mid-speech pauses, speech disfluencies and repaired utterances. It also included the 2 second pause detection times of the system, roughly adding 2-3 minutes to the translation. We could not reasonably expect the translation to be similar in size to the original.

Regarding quality, while we are aware that BLEU scores (Papineni et al., 2001) cannot be used for evaluating the absolute quality, we nevertheless tried to get an impression of the quality by running a single-reference BLEU against the translations.

### 3.3 Problems and improvements

What caused the short durations and low translation scores?

During the experiment, we noted that participants were struggling to translate parts of the interview. After transcribing and analysing the translations (cf. section 5.2), we discovered that many sentences were simply not translated at all, or only

<sup>3</sup><http://www.youtube.com/watch?v=iEofKzw7ZUg>

partially translated. Out of 85 sentences, the participants on average had not fully translated 36.3 sentences (Table 1).

Using our observations of participants and their BLEU scores, we analysed the problems with the approach. The original recording quality was too low, leading to many missing sentences. This in turn resulted in very low BLEU scores.

<i>Particip.</i>	A	B	C	D	E	F	G
<i>Missing</i>	36	32	35	38	41	40	32
<i>BLEU</i>	6.6	11.3	7.5	9.1	13.5	11.8	10.9

Table 1: Missing sentences and BLEU scores.

The participants remarked that hearing the interview for the first time was distracting: Jobim was a popular Brazilian musician who had a gift for storytelling. Participants simply got carried away by the story itself.

This feedback resulted in the following improvements. To mitigate problems with the missing context, we added an additional step to the procedure: before translating, participants would listen once to the entire recording. To avoid problems with poor audio quality, we decided to use a more recent recording which was of perfect audio quality and upgrade to slightly more expensive, but still entry-level HTC phones.<sup>4</sup>

## 4 Experiment 2

### 4.1 Subjects and Materials

Ten native speakers of Brazilian Portuguese, aged 20 to 32, from all areas of Brazil participated. One of the participants was a professional interpreter with a NATII accreditation.<sup>5</sup>

All had achieved a TOEFL score of at least 90 points,<sup>6</sup> the requirement to study at the University of Melbourne. They had at least four years of English lessons in high school. Most had only arrived recently and had two or more months of recent experience in speaking the language. Some have had more intensive exposure to English.

We used a high quality audio recording of a recent interview by Celsinho Cotrim with the first Brazilian female judge, Luislinda Valois<sup>7</sup>. The speakers are from the state of Bahia, speak relatively clearly and with a more neutral accent. This

<sup>4</sup>Priced at US\$ 160.

<sup>5</sup><http://www.naati.com.au/accreditation.html>

<sup>6</sup>TOEFL scores, <http://www.ets.org/toefl/ibt/scores>

<sup>7</sup><http://www.youtube.com/watch?v=oYK6uoyNGqA>

is representative of a realistic recording from the Aikuma system, data from a language archive, or a linguistic field recording.

The recording is spoken in Portuguese, has a total duration of 5:06 min and contains 90 sentences or phrases and 806 words in total. We selected this interview for various reasons: the content is of moderate complexity; the recording contains dialogue; the two speakers are not of the same gender, making it easier to distinguish their voices.

Some expressions can not be translated literally but have to be translated idiomatically. One example of this is the Brazilian idiom: ‘*Meus pais nunca abriram o mão do educação*’, literally ‘My parents would never open the hand of education’, which means ‘My parents would never drop education’.

### 4.2 Method

Given the feedback in the pilot experiments, we improved the process as follows:

For the training run, we used the same recording as used in the pilot experiment. We also demonstrated the newly introduced concept of removing the phone to stop playback if they needed to re-hear a particular segment. Removing the phone and putting it back on the ear would rewind the recording to the beginning of the last segment. During training, as soon as they seemed to have grasped the concept of translation, we stopped the training. On average, this took 1-2 minutes.

To provide context for the following translation, we asked the participants to listen to the entire original recording once, without performing any translation. None of the translators noted having problems understanding the audio or content.

We then instructed the participants to translate the original carefully without omitting any content. In case they encountered words where they did not know the English translation, we asked them to simply repeat the Portuguese word. If the English translation for a given word or expression was not known, we asked them to paraphrase. We instructed them to decide themselves where to segment the text, we specifically did not tell them to segment on sentence boundaries. Participants were then asked to translate the original recording a second time.

## 5 Results

We obtained 20 oral English translations, two per speaker, of the same Brazilian Portuguese source recording. All of these translations were carefully transcribed. From the audio recordings, we extracted a few key metrics from the recordings themselves.

### 5.1 Efficiency

To measure efficiency, we calculated the total time it took to listen to the original plus the time to translate it twice. The silences that are necessary for the interface to work are included in the duration of the task. Translation of the 5:06 min original took on average 15:39 min, a factor of roughly 3. In total, including preparation and listening, the process took slightly more than 35 minutes on average (Figure 2).

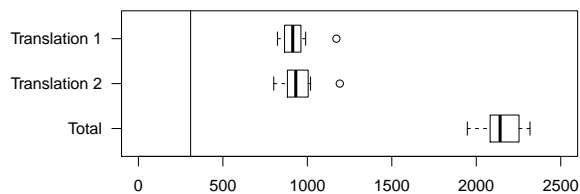


Figure 2: Durations of the recordings (s). The vertical line denotes the duration of the original.

### 5.2 Quality

#### 5.2.1 Preparation of recorded translations

To analyse and compare the translations, we prepared transcriptions of the original Brazilian Portuguese audio and all 20 English translations.

To evaluate translation quality, we used the Human-targeted Translation Error Rate (HTER), which requires a comparison between the resulting hypothesis translations and a number of reference translations (Snover et al., 2006). For this purpose, we prepared a parallel translation (Table 2). Due to speech disfluencies and repaired utterances, such as ‘um’ and ‘green I mean blue’ (Levelt, 1983), the process itself, and varying sentence segmentation the resulting transcriptions needed to be processed further for evaluation.

#### 5.2.2 HTER

HTER uses human annotators to create a specific targeted reference sentence for each translated hypothesis sentence. Each hypothesis sentence is edited by a bilingual editor until it is fluent and

Brazilian Portuguese	English
Eu sou mulher mais feliz do mundo	I am the happiest woman in the world
No importa por que	It does not matter why
Mas eu acho que sou	But I think I am

Table 2: Example reference translation.

has the same meaning as the source sentence.<sup>8</sup> As a targeted reference sentence has to be created for every hypothesis sentence, HTER is very resource intensive.

As we did not have the necessary resources to perform this analysis on all recordings (100 hours for 22 translations), we selected three example translations based on their BLEU scores (not mentioned) the lowest and highest result, and the expert’s translation.

For each of the selected translations, a bilingual annotator created targeted reference sentences. Then, we performed a TER on the six translations (Table 3).

Participant	Best	Worst	Expert
Run 1	0.16	0.23	0.18
Run 2	0.10	0.24	0.08

Table 3: Participant HTER scores in runs 1 and 2.

We found that the changes in HTER scores between translation runs agree with the BLEU score changes: both the “best” user and the expert receive an improved (numerically lower) score, while the “worst” user does not.

Regarding absolute scores, the expert comes out ahead of the “best” user. We assume that this is a result of the expert’s sophisticated use of English which was not present in the BLEU references.

## 6 Conclusions

We have investigated a new method for rapid translation of spoken language materials. The method can be used by amateur translators and offers a faster method for preserving endangered language data while there is still time. Our experiments indicate that the resulting translations are of sufficient quality to be useful in downstream NLP tasks.

<sup>8</sup>The standard HTER process only uses an untargeted reference in the target language to enable editing by monolingual editors.

## Acknowledgments

We gratefully acknowledge support of the Swiss National Science Foundation (Hanke) and the Australian Research Council (Bird).

## References

- Steven Abney and Steven Bird. 2010. The Human Language Project: building a universal corpus of the world's languages. In *Proceedings of the 48th Meeting of the Association for Computational Linguistics*, pages 88–97. Association for Computational Linguistics.
- Andrea Berez. 2007. Review of EUDICO linguistic annotator (ELAN). *Language Documentation & Conservation*, 1:283–289.
- Steven Bird. 2010. A scalable method for preserving oral literature from small languages. In *Proceedings of the 12th International Conference on Asia-Pacific Digital Libraries*, pages 5–14.
- Nikolaus Himmelmann. 1998. Documentary and descriptive linguistics. *Linguistics*, 36:1–34.
- Willem Levelt. 1983. Monitoring and self-repair in speech. *Cognition*, 14:41–104.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. *Science*, 22176:1–10.
- Will Reiman. 2010. Basic oral language documentation. *Language Documentation and Conservation*, 4:254–268.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231.
- Nicholas Thieberger. 2011. *The Oxford handbook of linguistic fieldwork*. Oxford University Press.
- Anthony Woodbury. 2007. On thick translation in linguistic documentation. *Language Documentation and Description*, 4:120–135.
- Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1220–1229.

# A Self-learning Template Approach for Recognizing Named Entities from Web Text

Qian Liu<sup>†‡</sup>, Bingyang Liu<sup>†‡</sup>, Dayong Wu<sup>‡</sup>, Yue Liu<sup>‡</sup>, Xueqi Cheng<sup>‡</sup>

<sup>†</sup>University of Chinese Academy of Sciences, Beijing, China

<sup>‡</sup>Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

{liuqian, liubingyang}@software.ict.ac.cn

{wudayong, liuyue, cxq}@ict.ac.cn

## Abstract

Recognizing Chinese Named entities from the Web is challenging, due to the lack of labeled data and differences between Chinese and English. We propose a semi-supervised approach which leverages seed entities and the large unlabeled data to learn templates. Some high-quality templates are generated iteratively to extract new named entities based on the model of quality metrics. Experimental results show that our approach significantly outperforms the baseline method and it is robust against the changes of the Web.

## 1 Introduction

Compared to English Named Entity Recognition (NER), Chinese NER is more difficult. For example, although capitalization plays an very important role in English NER, this language-specific feature is useless in Chinese NER. Moreover, the lack of space between words in Chinese often makes the work based on word segmentation (Sun et al., 2002) failure. Especially, there are a lot of new words and ambiguous words in the Web text, which increase the error of word segmentation. The loss of precision propagates to NER.

Currently, NER is mainly based on supervised models. Such models perform well in single domain (Wang, 2009; Du et al., 2010). Unfortunately, the data of the Web is open-domain and always changing. The performance of them degrades badly, since the distribution of the Web data is different from that of the training data. For instance, the average F1 score of the Stanford NER, which is trained on the CoNLL03 shared task data set and achieves state-of-the-art performance on that task, drops from 90.8% to 45.8% on tweets (Liu et al., 2011). Despite a high-quality training data set, which has the same distribution as the Web data

and covers all kinds of domains, can improve the performance of NER, as far as we know, there are no such labeled data. Furthermore, named entities change over the time, especially person names and company names. (Tsuchiya et al., 2009) shows that 20%~30% of named entity types are replaced with new ones every year in Mainichi Newspaper articles. Such change is even more obvious in the Web data and leads to the unreliable labeled data. Constantly annotating new data is time-consuming and expensive.

In this paper, we propose a semi-supervised approach that uses self-learning templates to solve above problems. Instead of annotating a massive amount of data, we leverage a small number of named entities and the large unlabeled data to discover new named entities that can not be identified using the training data. Experiments show that our approach raises the F1 from 75.9% to 88.6% on the Chinese Web data without retraining the existing model, and it is robust against the changes of the data. Since no language-specific knowledge is used, our approach can easily be extended to other languages.

## 2 Related Work

There have been many approaches proposed to solve the problem of the lack of annotated data. (Wu et al., 2009; Chiticariu et al., 2010) focus on domain adaptation, which aims to reuse the knowledge among different domains. (Ling and Weld, 2012; Rüd et al., 2011; Han and Zhao, 2010) leverage information from external knowledge sources, such as Wikipedia, WordNet and search engine, to compensate for the insufficient training data. Some work builds crowdsourcing services to label data by human. For example, Amazon Mechanical Turk<sup>1</sup> provides a platform to obtain data in various domains such as email

<sup>1</sup><https://www.mturk.com/mturk/welcome>



(Lawson et al., 2010), medicine (Yetisgen-Yildiz et al., 2010) and Twitter (Finin et al., 2010).

The work based on context templates is more closely related to our approach. (Etzioni et al., 2005) extracts named entities via domain-specific templates, which are learned from predefined templates. (Whitelaw et al., 2008) builds training data utilizing templates generated by millions of seeds. Although context templates are used to improve the perform of NER in our approach, they are learned automatically from the unlabeled data and we only use several seed entities.

### 3 Approach based on self-learning Templates

#### 3.1 Overall Framework of Our Approach

The main idea of our approach is that learning the high-quality templates in the bootstrapping process.

The details are shown in Algorithm 1, where the pair  $\langle name, type \rangle$  represents an entity, named  $name$ , in class  $type$ . # denotes a placeholder for entity. The substring, like  $t_{s-2}t_{s-1}name_s^{(i)}t_{s+1}t_{s+2}$ , denotes the  $i^{th}$  NE in the set of seed entities and its context of four tokens long. First, for each entity  $e$  in  $E_{seed}$ , we find sentences containing  $e$  and create a temporary set of templates. Second, for each candidate template, we relocate all possible named entities  $E_{temp}$  in  $C_{corpus}$ . Third, computing the quality of templates and adding the high-scoring ones into template set  $TS_{template}$ . Lastly, we compute the confidence of candidate entities that are generated in above process, and remove the entities whose confidence are below the threshold from the set of entities. The value of threshold will be discussed in Section 4.

#### 3.2 Features of Template

Given a candidate template, we define three statistical features to measure the quality of it.

**effectiveness ( $f_1$ ):** This feature reflects whether a candidate template is prone to mistakes. We assume that tokens outside of  $E_{seed}$  are not named entities. It is a reasonable assumption in practice, because the loss caused by assumption will become lower and lower with the increase of  $E_{seed}$ . The effectiveness is calculated as

$$f_1(T, c) = p(e|T_c) \cdot p(T_c) = \frac{\#(\text{correct } e|T_c)}{\sum_i \#(e|T_c^{(i)})} \quad (1)$$

#### Algorithm 1 Framework of Templates Learning

---

**Input:** a set of NEs:  $E_{seed} = \{ \langle name, type \rangle \}$ ; unlabeled web pages:  $C_{corpus}$

**Output:**  $E_{seed}; TS_{template}$

- 1: Initialize  $C_{corpus}: \phi$
- 2: Initialize  $TS_{candidate}: \phi$
- 3: Initialize  $TS_{template}: \phi$
- 4: **while**  $E_{seed}$  keep growing (or below the predefined number of loops) **do**
- 5:   **for** each  $entity^{(i)} = \langle name^{(i)}, type^{(i)} \rangle \in E_{seed}$  **do**
- 6:     Add all sentences containing  $name^{(i)}$  to  $C_{corpus}$
- 7:     Create templates  $\langle t_{-2}t_{-1}\#t_{+1}t_{+2}, type \rangle$  when the substring  $t_{s-2}t_{s-1}name_s^{(i)}t_{s+1}t_{s+2}$  belongs to some sentence in  $C_{corpus}$  and add them to  $TS_{candidate}$
- 8:   **end for**
- 9:   **for** each candidate template
- 10:      $T^{(i)} = \langle t_{-2}t_{-1}\#t_{+1}t_{+2}, type \rangle \in TS_{candidate}$  **do**
- 11:       Extract all matching tokens  $\langle token, type \rangle$  while the substring  $t_{-2}t_{-1}token_{+1}t_{+2}$  belongs to some sentence in  $C_{corpus}$  and add them to  $E_{temp}$
- 12:     **end for**
- 13:     **for** each template  $T^{(i)} \in TS_{candidate}$  **do**
- 14:       Calculate the score  $(T^{(i)}, score)$
- 15:        $= evaluation(TS_{candidate}|C_{corpus}, E_{seed}, E_{temp}, )$
- 16:       **if**  $score > \delta$  **then**
- 17:         Add  $T^{(i)}$  to  $TS_{template}$
- 18:       **end if**
- 19:     **end for**
- 20:     Find new candidate NEs  $E_{candidate}$  using  $TS_{template}$  from the remaining unlabeled data
- 21:     Select high-quality NEs:
- 22:      $E'_{seed} = filter(E_{candidate}|TS_{template})$
- 23:     Update:  $E_{seed} = E_{seed} \cup E'_{seed}$
- 24: **end while**

---

where  $\#(e|T_c^{(i)})$  denotes the number of correct entities extracted by the  $i^{th}$  template in class  $c$ .

**discrimination ( $f_2$ ):** This feature is used for measuring how close a candidate template is related to a class. The value is computed as

$$f_2(T, c) = tf(T, c) \cdot \left( 1 + \log \frac{\#C}{1 + \#c_j} \right) \quad (2)$$

where  $tf(T, c)$  denotes the normalized frequency of template, that is divided by the maximum frequency,  $\#C$  is the number of classes in  $E_{seed}$ , and  $\#c_j$  is the number of classes that template  $T$  appears.

**diversity ( $f_3$ ):** The more different and correct NEs in a class extracted by template  $T$ , the more likely other good NEs within the same class will be extracted by it. This feature is computed as

$$f_3(T, c) = \frac{\#\{(\text{correct } e|T_c) \wedge (\text{different } e|T_c)\}}{\#(\text{correct } e|T_c)} \quad (3)$$

#### 3.3 Quality Metrics Model of Templates

Since the proposed features are not independent of each other, we propose an approach in which the value of a feature is adapted according to the values of other features. Although it is similar to (Wei et al., 2010), we improve it by only updating a part of templates to reduce the computational cost.

Formally, given a set of candidate templates  $TS_{candidate} = \{T^{(1)}, T^{(2)}, \dots, T^{(n)}\} \subset R^m$ , let  $f_k : TS_{candidate} \rightarrow R$  denote the ranking function on the  $k^{th}$  feature, where  $f_k \in F = \{f_1, f_2, f_3\}$ . Our goal is to combine all features to produce ranking list that are better than any individual feature and then return the templates with high ranking scores.

Following the traditional manifold ranking process (Zhou et al., 2004) with one ranking function. 1) Defining the similarity matrix  $W$  on the template set  $TS_{candidate}$ :  $W_{ij} = similarity(T^{(i)}, T^{(j)})$ . 2) Symmetrically normalizing  $W$  by  $S = D^{-1/2}WD^{-1/2}$  in which  $D$  is the diagonal matrix with (i, i)-element equal to the sum of the  $i^{th}$  row of  $W$ . 3) Iterating  $F(t+1) = \alpha SF(t) + (1-\alpha)F(0)$  until a global stable state, where  $\alpha$  is trade-off parameter in (0, 1),  $F(0)$  denotes the initial ranking results and  $F(t)$  denotes the ranking results of the  $t^{th}$  round.

For two ranking functions  $f_1$  and  $f_2$ , the ranking score of  $f_1$  will be changed after combining the ranking score of  $f_2$ . Considering the cost of consistency both the ranking results in initial  $f_1$  and the feedback from  $f_2$ , we define the cost function caused by refining  $f_1$  with  $f_2$  in the  $(t+1)^{th}$  round iteration as

$$\varphi(f_1|f_2) = \frac{1}{2} \left( \sum_{i,j=1}^n W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} f_1^{(t+1)}(T^{(i)}) - \frac{1}{\sqrt{D_{jj}}} f_2^{(t)}(T^{(j)}) \right\|^2 + \mu \sum_{i=1}^n \left\| f_1^{(t+1)}(T^{(i)}) - f_1^{(0)}(T^{(i)}) \right\|^2 \right) \quad (4)$$

where  $f_1^{(0)}$  denotes the initial ranking scores of  $f_1$ . Let the best refined ranking score is  $f^*$ , we have

$$\frac{\partial}{\partial f_1} \varphi(f_1|f_2) |_{f_1=f^*} = f^* - S \cdot f_2^{(t)} + u(f_1^* - f_1^{(0)}) = 0 \quad (5)$$

$$f^* = \frac{1}{1+\mu} \cdot S \cdot f_2^{(t)} + \frac{\mu}{1+\mu} \cdot f_1^{(0)} \quad (6)$$

Let  $\alpha = \frac{1}{1+\mu}$ , then we have

$$f^* = \alpha \cdot S \cdot f_2^{(t)} + (1-\alpha) \cdot f_1^{(0)} \quad (7)$$

Therefore, we can iteratively compute the ranking scores in the  $(t+1)^{th}$  round to find the best  $f^*$  shown as follows, which is proven to be convergent (Wei et al., 2010).

$$f_1^{(t+1)} = \alpha \cdot S \cdot f_2^{(t)} + (1-\alpha) \cdot f_1^{(0)} \quad (8)$$

In our case, due to the templates with low ranking scores on  $f_2$  are helpless to refine the results

ranked on  $f_1$ . We omit the templates in  $f_2$  that fall below the threshold and feedback the rest templates, signaled by  $Top-f_2$ , to  $f_1$ . Then, the normalized matrix  $S$  can be simplified as a block matrix  $\begin{pmatrix} S_{Top-f_2} & 0 \\ 0 & 0 \end{pmatrix}$ . The equation above can be rewritten as

$$f_1^{(t+1)}(Top-f_2) = \alpha \cdot S_{Top-f_2} \cdot f_2^{(t)}(Top-f_2) + (1-\alpha) \cdot f_1^{(0)}(Top-f_2) \quad (9)$$

Moreover, if we only improve the identical templates, the similarity matrix  $W_{Top-f_2}$  and normalized matrix  $S_{Top-f_2}$  degrade to identity matrices, the final iteration equation is

$$f_1^{(t+1)}(Top-f_2) = \alpha \cdot f_2^{(t)}(Top-f_2) + (1-\alpha) \cdot f_1^{(0)}(Top-f_2) \quad (10)$$

### 3.4 Confidence of Candidate NEs

There may be still noisy in the set of candidate NEs despite using the high-quality templates to find new entities. Therefore, we also need to filter out the non-NEs for the further processing to insure the high accuracy. This section corresponds to the line 19 of Algorithm 1. We utilize the pointwise mutual information (PMI) (Etzioni et al., 2005) to measure the closeness between extracted NEs and templates.

Given an extracted named entity  $e$  and a template  $T$ , the PMI score is computed as

$$PMI(e, T) = \frac{Hits(e+T)}{Hits(e)} \quad (11)$$

where  $Hits(\cdot)$  denotes the number of sentences searched in the whole unlabeled corpus.

The confidence of  $e$  extracted by template  $T_c$  belonging to class  $c$  can be expressed as

$$confidence(entity, c) = \frac{1}{\#T_c} \sum_i PMI(entity, T_c^{(i)}) \quad (12)$$

where  $\#T_c$  denotes the number of templates in class  $c$ .

## 4 Experiments

### 4.1 Data Set

We conducted experiments on a real data set collected from the Web, which is from August 1<sup>th</sup> 2012 to August 31<sup>th</sup> 2012<sup>2</sup>. The details of the data set are given in Table 1. Two fine-grained categories of *Person* are considered: *Singer* and *Athlete*. Note that the NER on fine-grained categories is more difficult than that on coarse-grained

<sup>2</sup>During the preprocessing step, HTML tags and ads are eliminated from the data.

categories such as *Person*, *Location* and *Organization*. We randomly selected 8,955 sentences and manually labeled them as test data, which contains 232 singers and 1,807 athletes. The labeled data is further split into two parts: one for training the baseline system and the other for testing. The test data set includes 65 singers and 406 athletes. We trained a linear CRF model (Lafferty et al., 2001) as the baseline using the BILOU scheme (Ratinov and Roth, 2009).

	News	Forum	Microblog
Number	1,087,926	420,278	49,037,301

Table 1: The composition of the data set.

## 4.2 Experimental Analysis

Table 2 gives some examples of learned templates. Using the learned templates and extracted named entities, an additional recognizer can be built very easily. It can discover some new named entities that are left out by models trained on labeled data. We performed experiments to make comparison between baseline system and AD\_NER system, which integrates additional recognizer into the CRF model. The results are shown in Table 3~5, where  $p$  is the threshold of confidence score of candidate entity.

Template	Q_score	Num. of NEs
<i>Singer</i>		
欢、#、庾(Huan, #, Geng)	0.9000	60
#演唱(sing)	0.8109	279
天后#, (diva)	0.6620	319
听着#的歌(listen to the song)	0.6120	9
演唱#的歌(sing a song)	0.5820	16
<i>Athlete</i>		
冠军#, (champion)	0.9059	2340
名将#, (famous athlete)	0.8711	481
选手#, (player)	0.8382	2440
战胜#夺冠(win)	0.7724	123
选手#在比(in a competition)	0.6423	274

Table 2: Examples of templates and their qualities (Q\_score) and the number of extracted named entities.

	Precision	Recall	F1
Baseline	93.1	41.5	57.4
AD_NER( $p = 0.001$ )	71.3	83.1	76.8
AD_NER( $p = 0.01$ )	75.4	88.4	<b>81.4</b>
AD_NER( $p = 0.1$ )	90.6	47.7	62.5

Table 3: Results on *Singer*.

	Precision	Recall	F1
Baseline	97.8	65.9	78.8
AD_NER( $p = 0.001$ )	79.9	91.5	85.3
AD_NER( $p = 0.01$ )	92.8	88.3	<b>90.5</b>
AD_NER( $p = 0.1$ )	93.9	71.4	81.1

Table 4: Results on *Athlete*.

	Precision	Recall	F1
Baseline	97.3	62.3	75.9
AD_NER( $p = 0.001$ )	76.2	90.0	82.5
AD_NER( $p = 0.01$ )	91.3	86.1	<b>88.6</b>
AD_NER( $p = 0.1$ )	93.2	67.8	78.5

Table 5: Overall experimental results.

From Table 3 and Table 4, we find the best performance of baseline on *Singer* is 57.4%, 21.4% lower than that on *Athlete*. This can be explained as the scale of labeled data impacts the performance of supervised method, because, in Table 1, the instances of *Singer* is not as sufficient as *Athlete*. However, the performance of our approach in the two categories remains stable. This illustrates that the large-scale unlabeled data is useful in the case of a lack of training data.

As shown in Table 5, the best F1 of AD\_NER is 88.6%, which is 12.7% higher than F1 of baseline. Although there is a somewhat loss of precision, we obtain a large number of named entities. Moreover, the cost of our approach is lower than automatic annotation, since we do not need to retrain the supervised model. It is more effective when labeled data is complex and hard to construct while unlabeled data is abundant and easy to access. In the practice, our proposed approach can easily remain up-to-date and extend the well-trained supervised model without fine-tuning or any human intervention.

We further find that the overall precision of AD\_NER with threshold at 0.1 is only 1.9% higher than that with threshold at 0.01, but the loss of recall is 18.3%. For this reason, the threshold can be set to 0.01.

## 5 Conclusion

In this paper, we propose an approach to build an additional named entity recognizer that can assist the existing supervised models. The experimental results on the real data set from the Web show that our method improves the F1 score from 75.9% to 88.6%.

## References

- Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1002–1012.
- Junwu Du, Zhimin Zhang, Jun Yan, Yan Cui, and Zheng Chen. 2010. Using search session context for named entity recognition in query. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 765–766.
- Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Un-supervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.
- Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 80–88.
- Xianpei Han and Jun Zhao. 2010. Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 50–59.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.
- Nolan Lawson, Kevin Eustice, Mike Perkowitz, and Meliha Yetisgen-Yildiz. 2010. Annotating large e-mail datasets for named entity recognition with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 71–79.
- Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence*.
- Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 359–367.
- Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.
- Stefan Rüd, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 965–975.
- Jian Sun, Jianfeng Gao, Lei Zhang, Ming Zhou, and Changning Huang. 2002. Chinese named entity identification using class-based language model. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7.
- Masatoshi Tsuchiya, Shoko Endo, and Seiichi Nakagawa. 2009. Analysis and robust extraction of changing named entities. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 161–167.
- Yefeng Wang. 2009. Annotating and recognising named entities in clinical notes. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 18–26.
- Furu Wei, Wenjie Li, and Shixia Liu. 2010. irank: A rank-learn-combine framework for unsupervised ensemble ranking. *Journal of the American Society for Information Science and Technology*, 61(6):1232–1243.
- Casey Whitelaw, Alex Kehlenbeck, Nemanja Petrovic, and Lyle Ungar. 2008. Web-scale named entity recognition. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 123–132.
- Dan Wu, Wee Sun Lee, Nan Ye, and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1523–1532.
- Meliha Yetisgen-Yildiz, Imre Solti, Fei Xia, and Scott Russell Halgrim. 2010. Preliminary experience with amazon’s mechanical turk for annotating medical named entities. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 180–183.
- Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. *Advances in neural information processing systems*, 16:321–328.

# Accurate Parallel Fragment Extraction from Quasi-Comparable Corpora using Alignment Model and Translation Lexicon

Chenhui Chu, Toshiaki Nakazawa, Sadao Kurohashi

Graduate School of Informatics, Kyoto University

Yoshida-honmachi, Sakyo-ku

Kyoto, 606-8501, Japan

{chu,nakazawa}@nlp.ist.i.kyoto-u.ac.jp kuro@i.kyoto-u.ac.jp

## Abstract

Although parallel sentences rarely exist in quasi-comparable corpora, there could be parallel fragments that are also helpful for statistical machine translation (SMT). Previous studies cannot accurately extract parallel fragments from quasi-comparable corpora. To solve this problem, we propose an accurate parallel fragment extraction system that uses an alignment model to locate the parallel fragment candidates, and uses an accurate lexicon filter to identify the truly parallel ones. Experimental results indicate that our system can accurately extract parallel fragments, and our proposed method significantly outperforms a state-of-the-art approach. Furthermore, we investigate the factors that may affect the performance of our system in detail.

## 1 Introduction

In statistical machine translation (SMT) (Brown et al., 1993; Koehn et al., 2007), since translation knowledge is acquired from parallel data, the quality and quantity of parallel data are crucial. However, except for a few language pairs, such as English-French, English-Arabic, English-Chinese and several European language pairs, parallel data remains a scarce resource. As non-parallel corpora are far more available, extracting parallel data from non-parallel corpora is an attractive research field.

Most previous studies focus on extracting parallel sentences from comparable corpora (Zhao and Vogel, 2002; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005; Tillmann, 2009; Smith et al., 2010; Abdul-Rauf and Schwenk, 2011). Quasi-comparable corpora that contain far more disparate very-non-parallel bilingual docu-

ments that could either be on the same topic (in-topic) or not (out-topic) (Fung and Cheung, 2004), are available in far larger quantities than comparable corpora. In quasi-comparable corpora, there are few or no parallel sentences. However, there could be parallel fragments in comparable sentences that are also helpful for SMT.

Previous studies for parallel fragment extraction from comparable sentences have the problem that they cannot extract parallel fragments accurately. Some studies extract parallel fragments relying on a probabilistic translation lexicon estimated on an external parallel corpus. They locate the source and target fragments independently, making the extracted fragments unreliable (Munteanu and Marcu, 2006). Some studies develop alignment models for comparable sentences to extract parallel fragments (Quirk et al., 2007). Because the comparable sentences are quite noisy, the extracted fragments are not accurate.

In this paper, we propose an accurate parallel fragment extraction system. We locate parallel fragment candidates using an alignment model, and use an accurate lexicon filter to identify the truly parallel ones. Experimental results on Chinese-Japanese corpora show that our proposed method significantly outperforms a state-of-the-art approach, which indicate the effectiveness of our parallel fragment extraction system. Moreover, we investigate the factors that may affect the performance of our system in detail.

## 2 Related Work

(Munteanu and Marcu, 2006) is the first attempt to extract parallel fragments from comparable sentences. They extract sub-sentential parallel fragments by using a Log-Likelihood-Ratio (LLR) lexicon estimated on an external parallel corpus and a smoothing filter. They show the effectiveness of fragment extraction for SMT. This study has the drawback that they do not locate the source

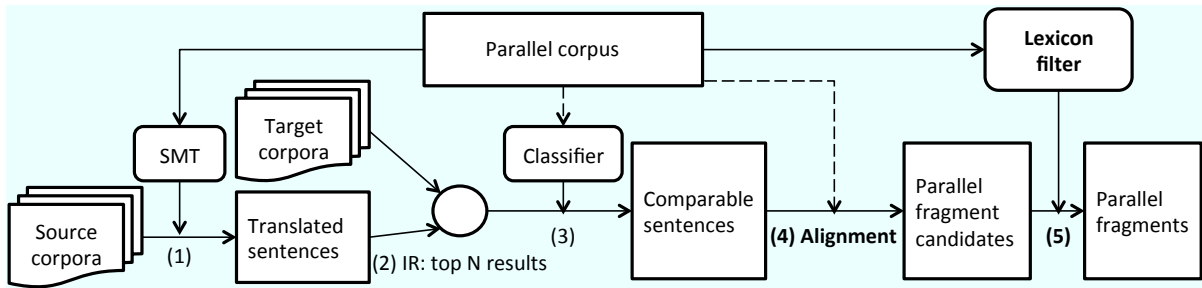


Figure 1: Parallel fragment extraction system.

and target fragments simultaneously, which cannot guarantee that the extracted fragments are translations of each other. We solve this problem by using an alignment model to locate the source and target fragments simultaneously.

Quirk et al. (2007) introduce two generative alignment models for extracting parallel fragments from comparable sentences. However, the extracted fragments slightly decrease MT performance when appending them to in-domain training data. We think the reason is that because the comparable sentences are quite noisy, the alignment models cannot accurately extract parallel fragments. To solve this problem we only use alignment models for parallel fragment candidate detection, and use an accurate lexicon filter to guarantee the accuracy of the extracted parallel fragments.

Besides the above studies, there are some other efforts. Hewavitharana and Vogel (2011) propose a method that calculates both the inside and outside probabilities for fragments in a comparable sentence pair, and show that the context of the sentence helps fragment extraction. However, the proposed method only can be efficient in a controlled manner that supposes the source fragment was known, and search for the target fragment. Another study uses a syntax-based alignment model to extract parallel fragments from noisy parallel data (Riesa and Marcu, 2012). Since their method is designed for noisy parallel data, we believe that the method cannot accurately extract parallel fragments from comparable sentences.

### 3 Proposed Method

#### 3.1 System Overview

Figure 1 shows an overview of our parallel fragment extraction system. We first apply comparable sentence extraction using a combination method

of (Abdul-Rauf and Schwenk, 2011) (1)(2) and (Munteanu and Marcu, 2005) (3), which were originally used for extracting parallel sentences from comparable corpora. We translate the source sentences to target language with a SMT system trained on a parallel corpus (1). Then we use the translated sentences as queries for IR. We retrieve the top 10 target documents for each source sentence using Indri<sup>1</sup>, and use all sentences in the documents as comparable sentence candidates (2). Next, we identify the comparable sentences from the candidates using a classifier trained on a part of a parallel corpus<sup>2</sup> following (Munteanu and Marcu, 2005) (3).

As the noise in comparable sentences will decrease MT performance, we further apply parallel fragment extraction. We apply two steps to accurately extract parallel fragments. We first detect parallel fragment candidates using bidirectional IBM models (Brown et al., 1993) with symmetrization heuristics (Koehn et al., 2007) (4). The generative alignment models proposed by Quirk et al. (2007) may be more efficient for parallel fragment candidate detection, we leave this for future work. Then we filter the candidates with probabilistic translation lexicon to produce accurate results (5). We present the details of our proposed method in following sections.

#### 3.2 A Brief Example

Figure 2 shows an example of comparable sentences extracted from Chinese-Japanese quasi-comparable corpora by our system. The alignment results are computed by IBM models. We notice that the truly parallel fragments “lead ion selective electrode” and “potentiometric titration method” are aligned, although there are some incorrectly aligned word pairs. We think this kind

<sup>1</sup><http://www.lemurproject.org/indri>

<sup>2</sup>In our experiments, we used 5k parallel sentences.

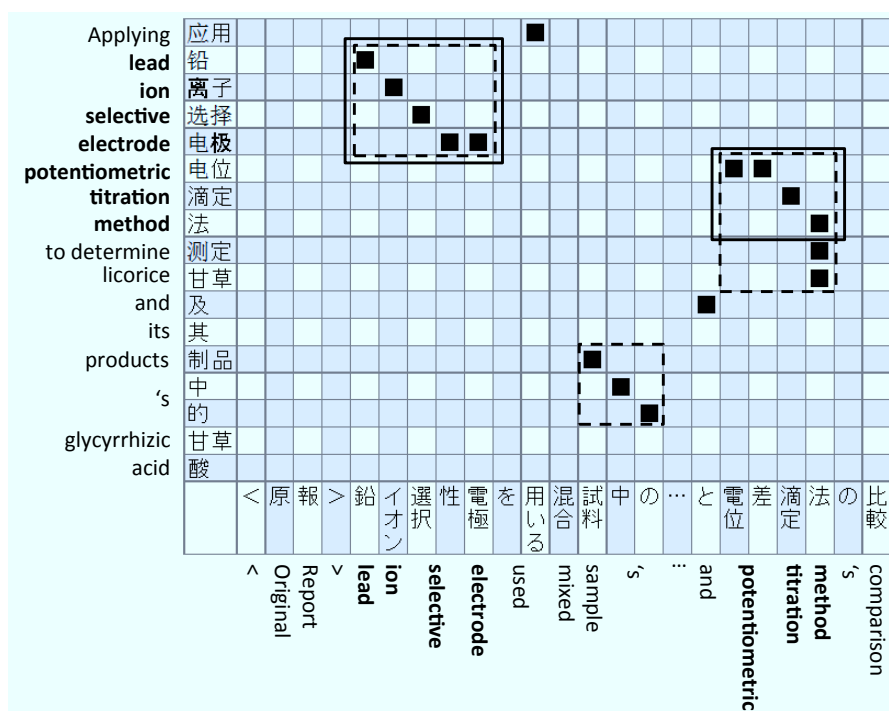


Figure 2: Example of comparable sentences with alignment results computed by IBM models (Parallel fragment candidates are in dashed rectangles, parallel fragments are in rectangles with solid line border).

of alignment information can be helpful for fragment extraction. What we need to do is develop a method to identify the true parallel fragments from the aligned fragments.

### 3.3 Parallel Fragment Candidate Detection

We treat the longest spans that have monotonic and non-null alignment as parallel fragment candidates. The reason we only consider monotonic ones is that based on our observation, ordering of IBM models on comparable sentences is unreliable. Quirk et al. (2007) also produce monotonic alignments in their generative model. Monotonic alignments are not sufficient for many language pairs. In the future, we plan to develop a method to deal with this problem. The non-null constraint can limit us from extracting incorrect fragments. Similar to previous studies, we are interested in fragment pairs with size greater than 3. Taking the comparable sentences in Figure 2 as an example, we will extract the fragments in dashed rectangles as parallel fragment candidates.

### 3.4 Lexicon-Based Filter

The parallel fragment candidates cannot be used directly, because many of them are still noisy as shown in Figure 2. Aiming to produce accurate

results, we use a lexicon-based filter. We filter a candidate parallel fragment pair with a probabilistic translation lexicon. The lexicon-pair may be extracted from a parallel corpus, or from comparable corpora using some state-of-the-art approaches such as (Vulić et al., 2011). In this study, we use the lexicon extracted from a parallel corpus. Different lexicons may have different effects for filtering. Here, we compare three types of lexicon. The first lexicon we use is the IBM Model 1 lexicon, which is obtained by running GIZA++<sup>3</sup> that implements sequential word-based statistical alignment model of IBM models.

The second lexicon we use is the LLR lexicon. Munteanu and Marcu (2006) show that the LLR lexicon performs better than the IBM Model 1 lexicon for parallel fragment extraction. One advantage of the LLR lexicon is that it can produce both positive and negative associations. Munteanu and Marcu (2006) develop a smoothing filter applying this advantage. We extract the LLR lexicon from a word-aligned parallel corpus using the same method as (Munteanu and Marcu, 2006).

The last lexicon we use is the SampLEX lexicon. Vulić and Moens (2012) propose an associative approach for lexicon extraction from par-

<sup>3</sup><http://code.google.com/p/giza-pp>

allel corpora that relies on the paradigm of data reduction. They extract translation pairs from many smaller sub-corpora that are randomly sampled from the original corpus, based on some frequency-based criteria of similarity. They show that their method outperforms IBM Model 1 and other associative methods such as LLR in terms of precision and F-measure. We extract SampleLEX lexicon from a parallel corpus using the same method as (Vulić and Moens, 2012).

Aiming to gain new knowledge that does not exist in the lexicon, we apply a smoothing filter similar to (Munteanu and Marcu, 2006). For each aligned word pair in the fragment candidates, we set scores to the words in two directions according to the extracted lexicon. If the aligned word pair exists in the lexicon, we set the corresponding translation probabilities as scores. For LLR lexicon, we use both positive and negative association values. If the aligned word pair does not exist in the lexicon, we set the scores in both directions to  $-1$ . There is the one exception that the aligned words are the same number, punctuation or abbreviation. In this case, we set the scores to 1 without considering the existence of the word pair in the lexicon. After this process, we get *initial scores* for the words in the fragment candidates in two directions.

We then apply an averaging filter to the *initial scores* to obtain *filtered scores* in both directions. The averaging filter sets the score of one word to the average score of several words around it. We think the words with initial positive scores are reliable, because they satisfy two strong constraints, namely alignment by IBM models and existence in the lexicon. Therefore, unlike (Munteanu and Marcu, 2006), we only apply the averaging filter to the words with negative scores. Moreover, we add another constraint that only filtering a word when both the left and right words around it have positive scores, which can further guarantee accuracy. For the number of words used for averaging, we used 5 (2 preceding words and 2 following words). The heuristics presented here produced good results on a development set.

Finally, we extract parallel fragments according to the *filtered scores*. We extract word aligned fragment pairs with continuous positive scores in both directions. Fragments with less than 3 words may be produced in this process, and we discard them like previous studies.

## 4 Experiments

In our experiments, we compared our proposed fragment extraction method with (Munteanu and Marcu, 2006). We manually evaluated the accuracy of the extracted fragments. Moreover, we used the extracted fragments as additional MT training data, and evaluated the effectiveness of the fragments for MT. We conducted experiments on Chinese–Japanese data. In all our experiments, we preprocessed the data by segmenting Chinese and Japanese sentences using a segmenter proposed by Chu et al. (2012) and JUMAN (Kurohashi et al., 1994) respectively.

### 4.1 Data

#### 4.1.1 Parallel Corpus

The parallel corpus we used is a scientific paper abstract corpus provided by JST<sup>4</sup> and NICT<sup>5</sup>. This corpus was created by the Japanese project “Development and Research of Chinese–Japanese Natural Language Processing Technology”, containing 680k sentences (18.2M Chinese and 21.8M Japanese tokens respectively). This corpus contains various domains such as chemistry, physics, biology and agriculture etc.

#### 4.1.2 Quasi-Comparable Corpora

The quasi-comparable corpora we used are scientific paper abstracts collected from academic websites. The Chinese corpora were collected from CNKI<sup>6</sup>, containing 420k sentences and 90k articles. The Japanese corpora were collected from CiNii<sup>7</sup> web portal, containing 5M sentences and 880k articles. Most articles in the Chinese corpora belong to the domain of chemistry, while the Japanese corpora contain various domains such as chemistry, physics and biology etc. Note that since the articles in these two websites were written by Chinese and Japanese researchers respectively, the collected corpora are very-non-parallel.

### 4.2 Extraction Experiments

We first applied sentence extraction on the quasi-comparable corpora using our system, and 30k comparable sentences of chemistry domain were extracted. We then applied fragment extraction on the extracted comparable sentences. We compared our proposed method with (Munteanu and

<sup>4</sup><http://www.jst.go.jp>

<sup>5</sup><http://www.nict.go.jp>

<sup>6</sup><http://www.cnki.net>

<sup>7</sup><http://ci.nii.ac.jp>



Method	# fragments	Average size (zh/ja)	Accuracy
Munteanu+, 2006	28.4k	20.36/21.39	(1%)
Only (IBM Model 1)	18.9k	4.03/4.14	80%
Only (LLR)	18.3k	4.00/4.14	<b>89%</b>
Only (SampLEX)	18.4k	3.96/4.05	87%
External (IBM Model 1)	28.7k	4.18/4.33	81%
External (LLR)	26.9k	4.17/4.33	85%
External (SampLEX)	28.0k	4.11/4.23	82%

Table 1: Fragment extraction results (Accuracy was manually evaluated on 100 fragments randomly selected from fragments extracted by different methods, based on the number of exact match).

Marcu, 2006). We applied word alignment using GIZA++. External parallel data might be helpful for alignment models to detect parallel fragment candidates from comparable sentences. Therefore, we compared two different settings to investigate the influence of external parallel data for alignment to our proposed method:

- **Only:** Only use the extracted comparable sentences.
- **External:** Use a small number of external parallel sentences together with the comparable sentences (In our experiment, we used chemistry domain data of the parallel corpus described in Section 4.1.1, containing 11k sentences).

We also compared IBM Model 1, LLR and SampLEX lexicon for filtering. All lexicons were extracted from the parallel corpus.

Table 1 shows the results for fragment extraction. We can see that the average size of fragments extracted by (Munteanu and Marcu, 2006) is unusually long, which is also reported in (Quirk et al., 2007). Our proposed method extracts shorter fragments. The number of extracted fragments and the average size are similar among the three lexicons when using the same alignment setting. Using the external parallel data for alignment extracts more fragments than only using the comparable sentences, and the average size is slightly larger. We think the reason is that the external parallel data is helpful to improve the recall of alignment for the parallel fragments in the comparable sentences, thus more parallel fragments will be detected.

To evaluate accuracy, we randomly selected 100 fragments extracted by the different methods. We manually evaluated the accuracy based on the number of exact match. Note that exact

match criteria has a bias against (Munteanu and Marcu, 2006), because their method extracts sub-sentential fragments which are quite long. We found that only one of the fragments extracted by “Munteanu+, 2006” is exact match, while for the remainder only partial matches are contained in long fragments. Our proposed method have a accuracy over 80%, while the remainder are partial matches. For the effects of different lexicons, LLR and SampLEX shows better performance than IBM Model 1 lexicon. We think the reason is the same one reported in previous studies that LLR and SampLEX lexicon are more accurate than IBM Model 1 lexicon. Also, LLR lexicon performs slightly better than SampLEX lexicon in this experiment. The accuracy of only using the comparable sentences for alignment are better than using the external parallel data, except for IBM Model 1 lexicon. We think the reason is that the external parallel data may have a bad effect on the precision of alignment for the parallel fragments in the comparable sentences.

### 4.3 Translation Experiments

We further conducted Chinese-to-Japanese translation experiments by appending the extracted fragments to a baseline system. For comparison, we also conducted translation experiments by appending the extracted comparable sentences. For decoding, we used the state-of-the-art phrase-based SMT toolkit Moses (Koehn et al., 2007) with default options, except for the distortion limit (6→20). The baseline system used the parallel corpus (680k sentences). We used another 368 and 367 sentences from the chemistry domain for tuning and testing respectively. We trained a 5-gram language model on the Japanese side of the parallel corpus using the SRILM toolkit<sup>8</sup>.

<sup>8</sup><http://www.speech.sri.com/projects/srilm>

System	BLEU
Baseline	38.64
+Sentences	39.16
+Munteanu+, 2006	38.87
+Only (IBM Model 1)	38.86
+Only (LLR)	39.27 <sup>†</sup>
+Only (SampLEX)	39.28 <sup>†</sup>
+External (IBM Model 1)	<b>39.63<sup>†*</sup></b>
+External (LLR)	39.22
+External (SampLEX)	39.40 <sup>†</sup>

Table 2: Results for Chinese-to-Japanese translation experiments (“<sup>†</sup>” and “<sup>‡</sup>” denotes the result is better than “Baseline” significantly at  $p < 0.05$  and  $p < 0.01$  respectively, “\*” denotes the result is better than “+Munteanu+, 2006” significantly at  $p < 0.05$ ).

Translation results evaluated on BLEU-4, are shown in Table 2. We can see that appending the extracted comparable sentences have a positive effect on translation quality. Adding the fragments extracted by (Munteanu and Marcu, 2006) has a negative impact, compared to appending the sentences. Our proposed method outperforms both “+sentences” and “Munteanu+, 2006”, which indicates the effectiveness of our proposed method for extracting useful parallel fragments for MT.

We compared the phrase tables produced by different methods to investigate the reason for different MT performance. We found that all the methods increased the size of phrase table, meaning that new phrases were acquired from the extracted data. However, the noise contained in the data extracted by “+sentences” and “Munteanu+, 2006” produced many noisy phrase pairs, which may decrease MT performance. Our proposed method extracted accurate parallel fragments, which led to correct new phrases. Among all the settings of our proposed method, “+External (IBM Model 1)” showed the best performance. The reason for this is that it extracted more correct parallel fragments than the other settings, thus more new phrase pairs were produced.

Surprisingly, the translation performance after appending the fragments extracted by our proposed method only using the comparable sentences for alignment shows comparable results when using LLR and SampLEX lexicon for filtering, compared to the ones using the external parallel data for alignment. We think the reason

is that the extracted fragments not only can produce new phrases, but also can improve the quality of phrase pairs extracted from the original parallel corpus. Because the fragments extracted only using the comparable sentences are more accurate than the ones using the external parallel data, they are more helpful to extract good phrase pairs from the original parallel corpus. This result indicates that external parallel data is not indispensable for the alignment model of our proposed method.

## 5 Conclusion and Future Work

In this paper, we proposed an accurate parallel fragment extraction system using alignment model together with translation lexicon. Experiments conducted on Chinese-Japanese data showed that our proposed method significantly outperforms a state-of-the-art approach and improves MT performance.

Our system can be improved in several aspects. Firstly, we only use IBM models for parallel fragment candidate detection, alignment models such as the ones proposed by (Quirk et al., 2007) could be more effective. Secondly, currently our proposed method cannot deal with ordering, an alignment model that is effective for ordering even on comparable sentences should be developed. Thirdly, although the experimental results indicate that external parallel data is not indispensable for the alignment model, we still use a parallel corpus for comparable sentence selection and lexicon filtering. An alternative method is constructing a large bilingual dictionary from comparable corpora, and use it for comparable sentence selection and lexicon filtering. Finally, although our proposed method is designed to be language and domain independent, the effectiveness for other language pairs and domains needs to be verified.

## References

- Sadaf Abdul-Rauf and Holger Schwenk. 2011. Parallel sentence generation from comparable corpora for improved smt. *Machine Translation*, 25(4):341–375.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Association for Computational Linguistics*, 19(2):263–312.
- Chenhui Chu, Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2012. Exploiting shared Chi-

- nese characters in Chinese word segmentation optimization for Chinese-Japanese machine translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT'12)*, Trento, Italy, May.
- Pascale Fung and Percy Cheung. 2004. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of Coling 2004*, pages 1051–1057, Geneva, Switzerland, Aug 23–Aug 27. COLING.
- Sanjika Hewavitharana and Stephan Vogel. 2011. Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 61–68, Portland, Oregon, 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.
- Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, December.
- Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 81–88, Sydney, Australia, July. Association for Computational Linguistics.
- Chris Quirk, Raghavendra Udupa U, and Arul Menezes. 2007. Generative models of noisy translations with applications to parallel fragment extraction. In *In Proceedings of MT Summit XI, European Association for Machine Translation*.
- Jason Riesa and Daniel Marcu. 2012. Automatic parallel fragment extraction from noisy data. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 538–542, Montréal, Canada, June. Association for Computational Linguistics.
- Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411, Los Angeles, California, June. Association for Computational Linguistics.
- Christoph Tillmann. 2009. A beam-search extraction algorithm for comparable data. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 225–228, Suntec, Singapore, August. Association for Computational Linguistics.
- Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79, Sapporo, Japan, July. Association for Computational Linguistics.
- Ivan Vulić and Marie-Francine Moens. 2012. Sub-corpora sampling with an application to bilingual lexicon extraction. In *Proceedings of COLING 2012*, pages 2721–2738, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Ivan Vulić, Wim De Smet, and Marie-Francine Moens. 2011. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 479–484, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Bing Zhao and Stephan Vogel. 2002. Adaptive parallel sentences mining from web abilingual news collections. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 745–748, Maebashi City, Japan. IEEE Computer Society.

# Meta-level Statistical Machine Translation

Sajad Ebrahimi<sup>†,\*</sup>, Kourosh Meshgi<sup>††</sup>, Shahram Khadivi<sup>†</sup>  
and Mohammad Ebrahim Shiri Ahmad Abady<sup>\*</sup>

<sup>†</sup>Human Language Technology Lab, Amirkabir University of Technology, Tehran, Iran

<sup>††</sup>Graduate School of Informatics, Kyoto University, Kyoto, Japan

<sup>\*</sup>Department of Computer Science, Amirkabir University of Technology, Tehran, Iran

ebrahimi.sajad@aut.ac.ir, meshgi-k@sys.i.kyoto-u.ac.jp

khadivi@aut.ac.ir, shiri@aut.ac.ir

## Abstract

We propose a simple and effective method to build a *meta-level* Statistical Machine Translation (SMT), called meta-SMT, for system combination. Our approach is based on the framework of *Stacked Generalization*, also known as *Stacking*, which is an ensemble learning algorithm, widely used in machine learning tasks. First, a collection of *base-level* SMTs is generated for obtaining a meta-level corpus. Then a meta-level SMT is trained on this corpus. In this paper we address the issue of how to adapt stacked generalization to SMT. We evaluate our approach on English-to-Persian machine translation. Experimental results show that our approach leads to significant improvements in translation quality over a phrase-based baseline by about 1.1 BLEU points.

## 1 Introduction

Currently, there exist a number of commercial and research Machine Translation (MT) systems, which are developed under different paradigms such as rule-based, example based, statistical machine translation, trained using different algorithms, e.g., phrase-based SMT, hierarchical phrase-based SMT, syntax-based SMT with different types and amounts of training data. With the emergence of these various structurally different systems, system combination methods have taken a great importance during the past few years.

There are several techniques for combining multiple SMT systems to achieve higher translation quality, e.g. sentence-level combination

(Hildebrand and Vogel, 2008) simply selects "the best" of the provided translations and phrase-level combination (Matusov et al., 2006; Rosti et al., 2007) can generate new translations differing from all original translations.

Most of the state-of-the-art SMT system combination methods require multiple SMT systems based on different models. Since it is not easy to have multiple SMT systems, in this work we focus on applying stacking algorithm on a single SMT system rather than multiple SMT systems.

We try to increase the performance of an SMT system by introducing a meta-level SMT which can learn how to decrease or modify translation errors on the translation outputs of original SMT system. This task is also known as automatic post-editing (APE) which is a well-studied topic in machine translation community (Simard et al., 2007a; Béchara et al., 2011). To do this, we use stacked generalization which is an ensemble learning algorithm

Ensemble Learning is a machine learning paradigm where multiple learners are trained to solve the same problem. An *ensemble* is viewed as a collection of learners which are usually called *base learners*. Base learners are usually generated from training data by a *base learning algorithm* which can be decision tree, neural network or other kinds of machine learning algorithms. Most ensemble methods use a single base learning algorithm to produce *homogeneous* base learners, but there are also some methods which use multiple learning algorithms to produce *heterogeneous* learners. The concept of ensembles appeared in classification literature has subsequently been studied in several frameworks, including *stacked generalization* (Wolpert, 1992),

*bagging* (Breiman, 1996b), *boosting* (Scharfire, 1990), *model averaging* (Perrone et al., 1993), *forecast combining* (Granger, 1989) and so on.

Previous work have tried to introduce some of these frameworks into SMT (Xiao et al., 2010), none of them adapt stacked generalization to SMT. While stacked generalization has been extensively investigated in machine learning, its adaption to SMT is not a trivial task. In this paper, we show how to make stacked generalization work for a single SMT system.

Stacked generalization is a general method of using a meta-level model to combine base-level models to achieve higher accuracy. However, this algorithm is introduced for combining multiple models, we focus our attention to utilize this algorithm in order to improve only a single SMT system. The basic idea of stacked generalization is to perform cross-validation on the base-level dataset in order to create a meta-level dataset. Then a meta-level model is trained on it. Finally, this system can generate better outputs than original system that is trained on the whole original dataset.

## 2 Background

Given a source sentence  $s$ , the goal of SMT is to find a target sentence  $t$  among all possible target strings  $t_1$ , that maximizes the probability:

$$t = \arg \max_{t_1} \{\text{pr}(t_1 | s)\}$$

Where  $\text{pr}(t_1 | s)$  is the probability that  $t_1$  is the translation of the given source string  $s$ . The target string  $t_1$  is a machine translation for  $s$ . In meta-SMT, a monolingual two-side corpus consists of these machine translations along with correct human translations. So, given a machinery output  $t$ , the goal of meta-SMT is to find a target sentence  $\hat{t}$ , that maximizes this probability:

$$\hat{t} = \arg \max_{t_2} \{\text{pr}(t_2 | t)\}$$

Where  $\text{pr}(t_2 | t)$  is the probability that  $t_2$  is the correct final translation of the given machine translated string  $t$  and both of  $t_2$  and  $t$  are in the same language. The target sentence  $\hat{t}$  is a final machine translation for  $s$ .

In the next section, we are going to describe stacked generalization for classification tasks and

in section 3, we present a general solution to adapting this algorithm to SMT.

### 2.1 Stacking for Classification

Wolpert (1992) introduced a novel approach for combining multiple classifiers, known as stacked generalization or stacking. The key idea is to learn a meta-level (or level-1) classifier based on the output of base-level (or level-0) classifiers, estimated via cross-validation as follows:

Define  $D = \{(x_i, y_i), i = 1, \dots, K\}$  as a data set, also referred to as level-0 data, where  $x_i$  is a feature vector representing the  $n$ th instance and  $y_i$  is the class value, and  $L_1 \dots L_N$  a set of different learning algorithms. During a  $J$ -fold cross-validation process,  $D$  is randomly split into  $J$  disjoint almost equal parts  $D_1, \dots, D_J$ . Define  $D^j$  and  $D \setminus D^j$  to be the test and training sets for  $j$ th fold of a  $J$ -fold cross-validation. At each  $j$ th fold,  $j = 1 \dots J$ , given the  $L_1 \dots L_N$  learning algorithms, we invoke each of them on the data in the training set  $D \setminus D^j$  to induce classifiers  $C_1(j) \dots C_N(j)$ . Then, these classifiers are applied to the test part  $D^j$ . The concatenated predictions of the induced classifiers on each feature vector  $x_i$  in  $D^j$ , together with the original class value  $y_i(x_i)$ , form a new  $MD^j$  of meta-level vectors.

At the end of the entire cross-validation process, the union  $MD = \cup_j MD^j, j = 1 \dots J$ , constitutes the full meta-level data set, also referred to as level-1 data, which is used for applying a learning algorithm  $L_M$  and inducing the meta-level classifier  $C_M$ . The learning algorithm that is employed at meta-level could be one of the  $L_1 \dots L_N$  or a different one. Finally, the learning algorithms are applied to the entire data set  $D$  inducing the final base-level classifiers  $C_1 \dots C_N$  to be used at runtime. In order to classify a new instance, the concatenated predictions of all base-level classifiers  $C_1 \dots C_N$  form a meta-level vector that is assigned a class value by the meta-level classifier  $C_M$ . In the next section we adapt this framework to SMT.

## 3 Adapting Stacking to SMT

Stacking is composed of two phases and we adapt it to SMT as follows:

First, a typical *SMT paradigm* is trained using  $J$ -fold cross-validation based on a bilingual cor-

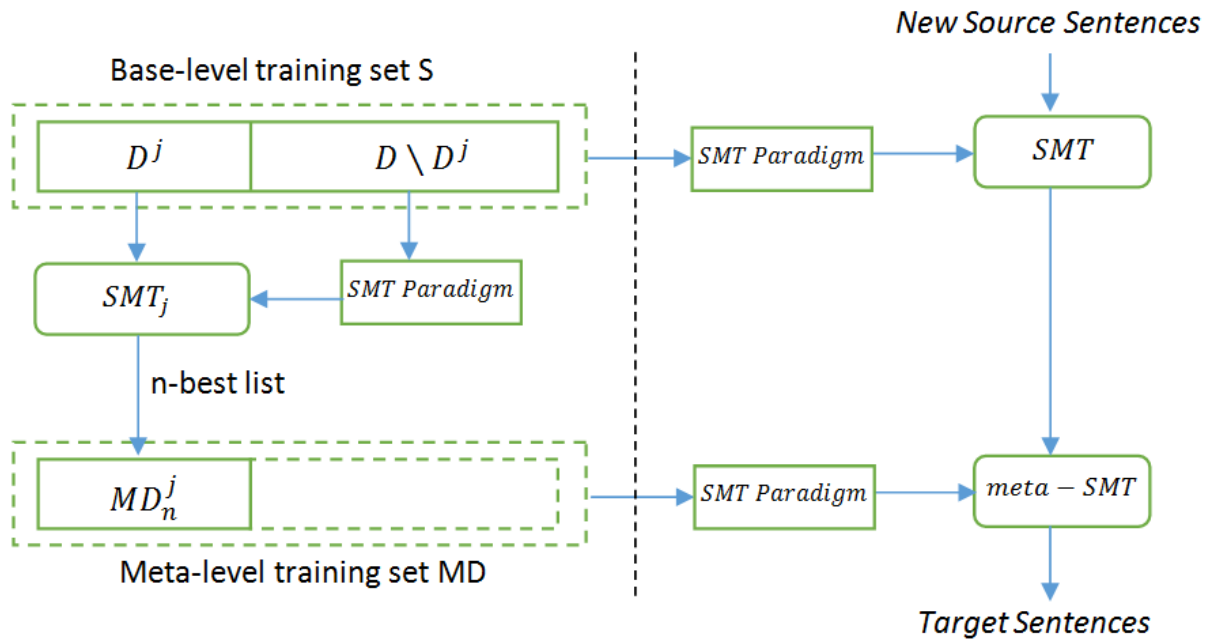


Figure 1. Stacking framework for SMT.

-pus. A popular setting of  $J$  is 5 and in this case it is called as *5-fold cross-validation*. We use this setting in our work. At the end of this step, 5 different systems are built based on 5 different training sets, called  $SMT_j, j = 1, \dots, 5$ . Then, the  $n$ -best outputs of these systems are collected to create a new corpus called meta-level corpus  $MD$ . For example, if we have a training corpus of  $N$  sentences and we use 3-best outputs of each system in cross-validation process, the effective size of the meta-level corpus will be  $3 \times N$ . In this new corpus, all the generated translations from the source sentences are paired to correct human translations.

Second, this corpus is used with another SMT paradigm -we call it *meta-SMT*- that could be identical to the SMT paradigm we used in cross-validation process or another SMT paradigm, in order to provide the final translation. In this algorithm, any SMT paradigm could be used in base-level and meta-level SMTs such as phrase-based SMT, hierarchical phrase-based SMT and syntax-based SMT. In this work, we utilize a phrase-based SMT for both base-level and meta-level. We use a phrase-based model for meta-level SMT, because we are supposed to improve a single SMT system. In addition, we train another phrase-based SMT based on full training set to produce target 1-best outputs and then use these outputs as input test set for meta-level SMT.

Figure 1 (left side) illustrates the cross-validation methodology, while Figure 1 (right

side) illustrates the stacking framework at runtime. Figure 2 also shows the algorithm in details.

### 3.1 Training base-level SMTs

After splitting the whole training corpus to separate training and test sets during cross-validation process, we train 5 phrase-based SMT systems on the training part and obtain the result of these systems on the corresponding test sets. We need these results for the next step.

### 3.2 Training meta-level SMTs

We gathered the  $n$ -best outputs of base-level SMTs on the corresponding test sets and built a meta-level corpus using these outputs along with correct human translations which was available from our original corpus. Then a meta-level SMT is trained on this corpus. We train our meta-SMT system on 10 meta-level corpus which is progressively created from  $n$ -best outputs of base-level systems,  $n = 1, \dots, 10$ ; i.e. each meta-level corpus that is created from  $n$ -best list also contains  $(1 \dots n - 1)$ -best list. In the results, we call these systems as meta-SMT (1-best) and meta-SMT (2-best) and so on.

### 3.3 Tuning meta-level SMTs

We must tune our meta-SMTs in a principled way. Similar to the training step, after splitting the whole tuning corpus to separate tuning and test sets during cross-validation process, we tune

---

**Input:** Training set  $D = \{(e_1, f_1), (e_2, f_2), \dots, (e_d, f_d)\}$ ;  
Tuning set  $T = \{(e'_1, f'_1), (e'_2, f'_2), \dots, (e'_t, f'_t)\}$ ;  
Test set  $S = \{(e''_1, f''_1), (e''_2, f''_2), \dots, (e''_s, f''_s)\}$ ;  
Base-level and Meta-Level SMT paradigms  $Bparadigm$  and  $Mparadigm$ , respectively.

**Process:**

- $baseline\_SMT = Bparadigm(D)$  % Train baseline SMT on the whole training set.
- $baseline\_SMT = baseline\_SMT(T)$  % Tune baseline SMT on the whole tuning set.
- $T_1 = baseline\_SMT(S)$  % Test baseline SMT on the original test set.
- $J$ -fold cross-validation: Divide the training and tuning sets into  $J$  roughly equal parts.
- $MD = \phi$ ; % Generate a new training set.
- $MT = \phi$ ; % Generate a new tuning set.

for  $j = 1, \dots, J$ :

$SMT_j = Bparadigm(D \setminus D^j)$  % Train base-level SMTs by their training parts  $D \setminus D^j$ .

$MD_n^j = SMT_j(D^j)$  % Test base-level SMTs on the corresponding test set to obtain  $n$ -best list.

$MD = MD \cup MD_n^j$  % Collect the outputs of base-level SMTs to create meta-level training corpus.

$SMT_j = SMT_j(T \setminus T^j)$  % Tune base-level SMTs by their tuning parts  $T \setminus T^j$ .

$MT^j = SMT_j(T^j)$  % Test base-level SMTs on the corresponding test set of tuning set.

$MT = MT \cup MT^j$  % Collect the outputs of base-level SMTs to create meta-level tuning corpus.

end;

$meta - SMT = Mparadigm(MD)$  % Train meta-level SMT by applying it to the new training corpus.

$meta - SMT = meta - SMT(MT)$  % Tune meta-level SMT by applying it to the new tuning corpus.

**Output:**  $T_2 = meta - SMT(T_1)$  % Test meta-SMT on the outputs of baseline SMT as test set.

---

Figure 2. Stacking algorithm adapted to SMT

5 base-level SMT systems on the tuning part and obtain the result of these systems on the corresponding test sets. Finally a meta-level development set is created by gathering these outputs paired with correct human translations to tune meta-level SMTs.

## 4 Experiments

### 4.1 Data

The corpus that is used for training and cross-validation process is Verbmobil project corpus which includes some tourists' conversations about time scheduling and appointment settings in German and English (Ney, 2000). Then a large part of English sentences are translated to Persian by human translators to build an English-Persian corpus (Bakhshaei et al., 2010). This dataset includes 23K lines in both sides, 249K and 216K words in Persian and English sides, respectively. We have chosen this corpus because it is small enough to perform cross-validation.

### 4.2 Experimental Setup

We use GIZA++ (Och and Ney, 2000) to perform the bi-directional word alignment between source and target side of each sentence pair. The final word alignment is generated using the *grow-diag-final-and* symmetrizing strategy. To speed up alignment, all the sentences with more than 80 words are removed. A 3-gram language model is trained on the target side of the bilingual data using the SRILM toolkit (Stolcke, 2002). The translation quality is evaluated in terms of case-insensitive BLEU metric.

We have run a phrase-based statistical machine translation with the Moses decoder (Koehn et al., 2007) to build baseline, base-level and meta-level SMTs.

We use MERT (Och, 2003) to tune the feature weights on the development data.

### 4.3 Evaluation

We investigate the effectiveness of our approach on improving a phrase-based SMT system. BLEU scores are computed on 250 test sentences

Type of SMT	Test set
<i>baseline SMT</i>	30.47
<i>meta-SMT (1-best)</i>	31.20
<i>meta-SMT (2-best)</i>	31.00
<i>meta-SMT (3-best)</i>	31.37
<b><i>meta-SMT (4-best)</i></b>	<b>31.49</b>
<b><i>meta-SMT (5-best)</i></b>	<b>31.41</b>
<i>meta-SMT (6-best)</i>	31.05
<i>meta-SMT (7-best)</i>	31.19
<b><i>meta-SMT (8-best)</i></b>	<b>31.40</b>
<i>meta-SMT (9-best)</i>	31.30
<b><i>meta-SMT (10-best)</i></b>	<b>31.54</b>

Table 1: BLEU (%) scores of baseline SMT and meta-SMTs on the Verbmobil test set.

with four reference translations. Table 1 shows the results of our approach against baseline SMT on the test set. We see that almost all meta-SMTs are achieved over 0.5 BLEU improvement on the test set. The biggest improvement is obtained by meta-SMT (10-best) by 1.07 BLEU improvement. Moreover, we see a similar behavior of our approach on the development set. Considering the results on the development set, meta-SMT (4-best) and meta-SMT (8-best) will be good choices for meta-level SMT.

While these results are very encouraging, we must investigate why this approach is helpful. We find that two factors possibly contribute to these results. first, performing cross-validation on the training set; second, and possibly more importantly, the re-optimization on the system. In order to verify whether the improvements are due to the cross-validation or re-optimizing, we perform two experiments. The first is to test this approach without any cross-validation process, but with the development set obtained from stacking. It means that all source side sentences of the training corpus are translated by using an SMT system that is trained on the bilingual same corpus. This experiment is referred to as Straight1 in the results. The second is to build meta-level SMTs tuned with a development set which is obtained directly from baseline SMT (i.e., without performing cross-validation on it). This experiment is referred to as Straight2 in the results.

A comparison between the results of the three settings (Stacking, Straight1 and Straight2) is shown in Figure 3. The figure shows BLEU curve on the test set, where the X-axis is the number of base-level outputs (n-best) that is used to create meta-level corpus for meta-SMT, and

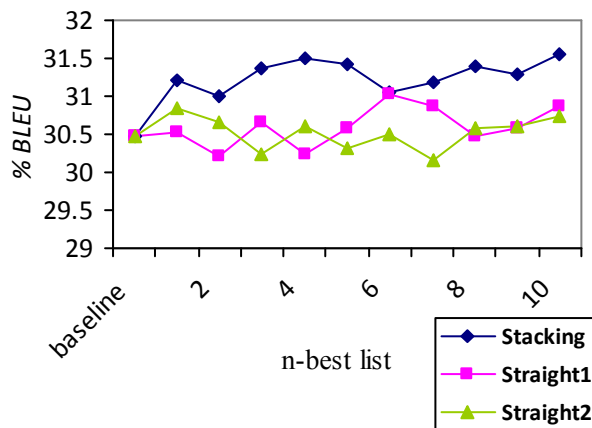


Figure 3. Comparison of Stacking, Straight1 and Straight2.

the Y-axis is the BLEU scores of the final meta-SMT calculated from each approach. After analyzing the results, it can be concluded that both factors, i.e., cross-validation and re-optimizing the system with the stacking-based development set, are important to outperform the baseline SMT system. Since use of both factors, consistently lead to the best results.

In all of the experiments, the size of the n-best list varies from 1 up to 10. The main reason for the upper limit is just that the experiments are very time consuming.

We conducted statistical significance tests using *paired bootstrap resampling* proposed by Koehn (2004) to measure the reliability of the conclusion that meta-SMTs are really better than baseline SMT. It is observed that all stacking-based meta-SMTs are really better than the baseline SMT in 99% of the times.

## 5 Related Work

Stacking is a machine learning (ML) algorithm that is a well-studied topic in the ML community (Wolpert, 1992; Breiman, 1996a), and has been successfully adapted in natural language processing and information retrieval, such as named entity recognition (Wu et al., 2003) and Information extraction (Sigletos et al., 2005).

There are also some researches on applying ensemble learning algorithms into SMT. Xiao et al. (2010) presented a general solution for adaptation of bagging and boosting to SMT. The results of their work showed that ensemble learning algorithms are promising in SMT.

Most other researches are in the statistical post-editing (SPE) techniques which have been used successfully to improve the output of Rule-



Based MT (RBMT) systems. Simard et al. (2007a), trained a “mono-lingual” Phrase-based SMT system (the *Portage* system) on the output of an RBMT system for the source side of the training set of the Phrase-based SMT system and the corresponding human translated (manually post-edited) reference. More recently, Béchara et al. (2011) designed a full phrase-based SMT pipeline that included a translation step and a post-editing step. The authors report significant improvements of 2 BLEU points for a French to English translation task, using a novel context aware approach. This method takes into account the source sentences during the post-editing process through a word-to-word alignment between the source words and the target words generated by the translation system.

As far as we are aware, the research presented in this paper is the first attempt to apply stacking algorithm to SMT with the configuration presented.

## 6 Conclusion and Future Work

We have presented a simple and effective approach to translation error modification by building a meta-level SMT using a meta-level corpus that is created from original corpus by cross validation. Experimental results showed that such a meta-SMT can fix many translation errors that occur in the baseline translations. The proposed method outperforms the baseline SMT on the same test set. We also believe that stacked generalization can be used to combine multiple SMT systems. As a future work, we have planned to develop a technique for combining multiple SMT systems using stacked generalization algorithm.

Moreover, we are running more tests with different language-pairs and larger corpora. We also have planned to use the confusion network methods on the input of meta-level SMTs, so that the meta-SMT can translate a confusion network built based on the n-best output of baseline SMT. As another future work, we will apply our framework under different SMT paradigms such as hierarchical phrase-based SMT and syntax-based SMT.

## Acknowledgments

We would like to thank Somayeh Bakhshaei and Abbas Masoumzadeh for their helpful discussions.

## References

- Almut Silja Hildebrand and Stephan Vogel. 2008. Combination of machine translation systems via hypothesis selection from combined n-best lists. *In Proc. of the 8th AMTA conference*, pages 254-261.
- Evegeny Matusov, Nicola Ueffing and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. *In Proc. of EACL 2006*, pages 33-40.
- Antti-Veikko Rosti, Spyros Matsoukas and Richard Schwartz. 2007. Improved word-level system combination for machine translation. *In Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, pages. 312-319.
- Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007a. Statistical phrase-based post-editing. *In NAACL-HLT*, pages 508-515
- Béchara, H., Y. Ma, and J. van Genabith. 2011. Post-editing for a statistical MT system. *In MT Summit XIII*, pages 308-315
- David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2): 241-259.
- Leo Breiman. 1996b. Bagging predictors. *Machine Learning*, 24(2):123-140.
- Robert E. Scharpire. 1990. The strength of weak learnability. *Machine Learning*, 5(2):197-227.
- Michael P. Perrone and Leao N. Cooper. 1993. When networks disagree: ensemble methods for hybrid neural networks. *Neural Networks for Speech and Image Processing*. Chapman-Hall, Chapter 10.
- Clive W.J Granger. 1989. Combining forecasts-twenty years later. *Journal of Forecasting*, 8(3):167-173.
- Tong Xiao, Jingbo Zhu, Muhua Zhu and Huizhen Wang. 2010. Boosting-based system combination for machine translation. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 739-748.
- Hermann Ney, Franz J. Och and Stephan Vogel. 2000. Statistical translation of spoken dialogues in the verbmobil system. *In Workshop on Multi-Lingual Speech Communication*, pages 69-74.
- Somayeh Bakhshaei, Shahram Khadivi and Noushin Riahi. 2010. Farsi-German statistical machine

- translation through bridge language. *Telecommunications (IST), 5th International Symposium on*, pages 557-561.
- Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. *In Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440-447.
- Andreas Stockle. 2002. SRILM - an extensible language modeling toolkit. *In Proc. of International Conference for Spoken Language Processing*, pages 901-904.
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *In Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*. Demo and Poster Sessions, pages. 177-180.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. *In Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*, pages 160-167.
- Philip Koehn, 2004. Statistical significance tests for machine translation evaluation. *In Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388-395.
- Leo Breiman. 1996a. Stacked regressions. *Machine Learning*, 24(1): 49-64.
- Dekai Wu, Grace Ngai and Marine Carpuat. A stacked, voted, stacked model for named entity recognition. *In Proc. of CoNLL-2003*, pages 200-203.
- Georgios Sigletos, Georgios Paliouras, Constantine D. Spyropoulos and Michali Hatzopoulos. 2005. Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6: 1751-1782.
- Tong Xiao, Jingbo Zhu and Tongran Liu. 2013. Bagging and boosting statistical machine translation systems. *Artificial Intelligence*. vol. 195, pages 496-527.
- Kai Ming Ting and Ian H. Witten. 1999. Issues in stacked generalization. *Journal of Artificial Intelligence Research (JAIR)*, 10: 271-289.
- Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC
- Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007b. Rule-based translation with statistical phrase-based post-editing. *In Proc. of WMT07*, pages 203-206

# Bayesian Induction of Bracketing Inversion Transduction Grammars

Markus SAERS Dekai WU

Human Language Technology Center  
Department of Computer Science and Engineering  
Hong Kong University of Science and Technology

{masaers|dekai}@cs.ust.hk

## Abstract

We present a novel approach to learning phrasal inversion transduction grammars via Bayesian MAP (maximum *a posteriori*) or information-theoretic MDL (minimum description length) model optimization so as to incorporate simultaneously the choices of model structure as well as parameters. In comparison to most current SMT approaches, the model learns phrase translation lexicons that (a) do not require enormous amounts of run-time memory, (b) contain significantly less redundancy, and (c) provide an obvious basis for generalization to abstract translation schemas. Model structure choice is biased by a description length prior, while parameter choice is driven by data likelihood biased by a parameter prior. The search over possible model structures is made feasible by a novel top-down rule segmenting heuristic which efficiently incorporates estimates of the posterior probabilities. Since the priors reward model parsimony, the learned grammar is very concise and still performs significantly better than the maximum likelihood driven bottom-up rule chunking baseline.

## 1 Introduction

We introduce a minimalist, unsupervised learning model that induces relatively clean, compact phrasal translation lexicons by employing a novel Bayesian approach that attempts to find the maximum *a posteriori* (MAP) or minimum description length (MDL) model. The approach iteratively segments the rules in a top-down fashion which allows for efficient estimation of the model prior and the likelihood of the data given a limited change in the model—it is, in other words, possible to gener-

ate a set of possible rule segmentations and compare them using the model posteriors or description length.

Our new approach differs from most other SMT approaches to unsupervised learning of phrasal translations, which (a) require enormous amounts of run-time memory, (b) contain a high degree of redundancy, and (c) do not provide an obvious basis for generalization to abstract translation schemas. The current state-of-the-art in SMT (Koehn *et al.*, 2003; Chiang, 2005) relies on long pipelines of mismatched learning models and heuristics. There is no way for latter stages of the pipeline to recover a mistake of omission made in an earlier stage, which forces the individual steps to massively overgenerate hypotheses. This typically manifests as massive redundancy in the phrasal lexicon, which causes significant overhead at run-time. The fact that it is even possible to improve the performance of a phrase-based direct translation system by tossing away most of the learned segmental translations (Johnson *et al.*, 2007) illustrates these deficiencies well. By staying within a single framework throughout training and testing, we do not have to overgenerate hypotheses—instead, we are able to evaluate their effect on the posterior model probability at the time they are proposed during learning. This cuts down the size of the phrasal lexicon significantly, and consequently saves the decoder a lot of run-time resources. The fact that we learn a phrasal inversion transduction grammar, or ITG (Wu, 1997) also means that the power to generalize and abstract over categories is built into the formalism (although we will not make use of this feature in this work).

A word as to the bigger-picture motivation for this line of inquiry may be necessary. By insisting on the fundamental machine learning principle of matching the training model to the testing model, we accept forfeiting the short term boost

in BLEU that is typically seen when embedding a learned ITG in the midst of the common heuristics employed in statistical machine translation. For example, Cherry and Lin (2007); Zhang *et al.* (2008); Blunsom *et al.* (2008, 2009); Haghighi *et al.* (2009); Saers and Wu (2009, 2011); Blunsom and Cohn (2010); Burkett *et al.* (2010); Riesa and Marcu (2010); Saers *et al.* (2010); Neubig *et al.* (2011, 2012) all plug some aspect of the ITGs they learn into training pipelines for existing, mismatched decoders, typically in the form of the word alignment that an ITG imposes on a parallel corpus as it is biparsed. Although this allows us to tap into the vast engineering efforts that have gone into tweaking existing decoders, it also prevents us from understanding the quality of the learned transduction grammar, whose characteristics become obscured by the many unrelated variables in the subsequent processing pipeline. Our own past work has also taken similar approaches, but it is not necessary to do so—instead, any ITG can be used for decoding by directly parsing with the input sentence as a hard constraint, as we do in this paper. The motivation for our present series of experiments is that as a field we are well served by tackling the fundamental questions as well, and not exclusively focusing on engineering short term incremental BLEU score boosts where the quality of an induced ITG itself is obscured because it is embedded within many other heuristic algorithms.

Bayesian approaches to grammar induction have a long history in computation linguistics. Starting with monolingual grammar induction (Chen, 1995; Stolcke and Omohundro, 1994), and moving on to transduction grammar induction (Blunsom *et al.*, 2008, 2009; Blunsom and Cohn, 2010; Neubig *et al.*, 2011, 2012). So far, the induced transduction grammar have only been used to derive Viterbi-style word alignments to feed into existing translation system, and there has been no evaluation of the grammars actually learned. In contrast, we directly evaluate the grammars that we induce.

Our algorithm for learning the structure of an ITG relies on segmenting known bilingual segments, starting with the sentence pairs of the training data, and continuing with the segments learned in this way. This is similar to the Recursive Alignment Model, or MAR (Vilar, 2005; Vilar and Vidal, 2005). Our method is, however, learning a full ITG, where MAR only learns a translation lex-

icon; furthermore, MAR is a discriminative model, whereas ours is a generative.

Transduction grammars can also be induced from treebanks instead of unannotated corpora, which cuts down the vast search space by enforcing additional, external constraints—taking it from the realm of unsupervised induction into the realm of supervised induction. This approach was pioneered by Galley *et al.* (2006) with numerous variants in subsequent research, usually referred to as **tree-to-tree**, **tree-to-string** and **string-to-tree**, depending on where the analyses are found in the training data. Our view on this line of research is that it complicates the learning process by adding external constraints that are bound to match the translation model poorly; grammarians of English should not be expected to care about its relationship to Chinese. It does, however, constitute a way to borrow nonterminal categories that help the translation model.

The work presented in this paper is related to our preliminary work with description length as learning objective (Saers *et al.*, 2013b). A key difference lies in the added parameter training, which facilitates a completely Bayesian interpretation. The present paper aims to be self-contained, by explaining the relationships throughout.

## 2 Background

In this section we briefly survey essential foundations for inversion transduction grammars and description length together with its Bayesian interpretation—in other words, what we search for, and how.

### 2.1 Inversion transduction grammars

Inversion transduction grammars, or ITGs (Wu, 1997), are an expressive yet efficient way to model translation. Much like context-free grammars (CFGs), they allow for sentences to be explained through composition of smaller units into larger units, but where CFGs are restricted to generate monolingual sentences, ITGs generate *pairs of sentences*—**transductions** rather than languages. Naturally, the components of different languages may have to be ordered differently, which means that transduction grammars need to handle these differences in order. Rather than allowing arbitrary reordering and pay the price of exponential time complexity, ITGs allow the only monotonically straight or inverted order of the productions,

which cuts the time complexity down to a manageable polynomial.

Formally, an ITG is a tuple  $\langle N, \Sigma, \Delta, R, S \rangle$ , where  $N$  is a finite nonempty set of nonterminal symbols,  $\Sigma$  is a finite set of terminal symbols in  $L_0$ ,  $\Delta$  is a finite set of terminal symbols in  $L_1$ ,  $R$  is a finite nonempty set of inversion transduction rules and  $S \in N$  is a designated start symbol. An inversion transduction rule is restricted to take one of the following forms:

$$S \rightarrow [A], A \rightarrow [\Psi^+], A \rightarrow \langle \Psi^+ \rangle$$

where  $S \in N$  is the start symbol,  $A \in N$  is a nonterminal symbol, and  $\Psi^+$  is a nonempty sequence of nonterminals and biterminals. A **biterminal** is a pair of symbol strings:  $\Sigma^* \times \Delta^*$ , where at least one of the strings have to be nonempty. The square and angled brackets signal straight and inverted order respectively. With straight order, both the  $L_0$  and the  $L_1$  productions are generated left-to-right, but with inverted order, the  $L_1$  production is generated right-to-left. The brackets are frequently left out when there is only one element on the right-hand side, which means that  $S \rightarrow [A]$  is shortened to  $S \rightarrow A$ .

Like CFGs, ITGs also have a 2-normal form, analogous to the Chomsky normal form for CFGs, where the rules are further restricted to only the following four forms:

$$S \rightarrow A, A \rightarrow [BC], A \rightarrow \langle BC \rangle, A \rightarrow e/f$$

where  $S \in N$  is the start symbol,  $A, B, C \in N$  are nonterminal symbols and  $e/f$  is a biterminal string.

## 2.2 MAP and MDL

Our approach to transduction grammar induction can be equivalently interpreted either from a Bayesian perspective as finding the grammar model  $\Phi$  with maximum *a posteriori* probability (MAP) given training data corpus  $D$ , or from a compression perspective as finding the model  $\Phi$  with minimum description length (MDL) needed to encode data  $D$  so we can transmit both the encoded data and the model needed to decode it using as few bits as possible.

In the MAP case, the goal is to find the model  $\Phi$  with the maximum posterior probability, given the data  $D$  and assuming a prior  $P(\Phi)$  over the space of models:

$$P(\Phi|D) = \frac{P(\Phi)P(D|\Phi)}{P(D)}$$

which gives the following search problem:

$$\operatorname{argmax}_{\Phi} P(\Phi|D) = \operatorname{argmax}_{\Phi} P(\Phi)P(D|\Phi)$$

since the data is fixed.

In the MDL case, the minimum description length principle is about compressing a corpus by finding the optimal balance between the size of a model,  $DL(\Phi)$ , and the size of some data given the model,  $DL(D|\Phi)$  (Solomonoff, 1959; Rissanen, 1983). In information theoretic terms, we encode the data with a model, and then transmit both the encoded data *and* the information needed to decode the data (the model) over a channel; the minimum description length is the minimum number of bits we can get away with sending over the channel. The encoded data can be interpreted as carrying the information necessary to disambiguate the uncertainties that the model has about the data. The model can *grow in size* and become *more certain* about the data, or it can *shrink in size* and become *more uncertain* about the data. Formally, description length (DL) is:

$$DL(\Phi, D) = DL(\Phi) + DL(D|\Phi)$$

which gives the following search problem:

$$\operatorname{argmin}_{\Phi} DL(\Phi, D) = \operatorname{argmin}_{\Phi} DL(\Phi) + DL(D|\Phi)$$

which is equivalent to the MAP search problem if we follow the Shannon (1948) lower bound on the number of bits required to encode a specific outcome of a random variable such that  $DL(\cdot) = -\lg P(\cdot)$  and conversely  $P(\cdot) = 2^{-DL(\cdot)}$ , since in that case the description length of the model  $DL(\Phi) = -\lg P(\Phi)$  and the description length of the data given the model  $DL(D|\Phi) = -\lg P(D|\Phi)$ . These two interchangeable views of the problem are complimentary; in our previous work on minimizing description length (Saers *et al.*, 2013a,b), we have used this equality to model the description length of the data given the model in terms of the probability of biparsing a parallel corpus with an ITG.

In Bayesian modeling, it is frequently useful to break out different aspects of the prior. For transduction grammars, we will break the prior into three aspects: the type of transduction grammar ( $\Phi_G$ ), the structure of the grammar (the specific set of rules conforming to the type,  $\Phi_S$ ), and the parameters of the grammar ( $\theta_{\Phi}$ ). The prior is thus

broken down such that:

$$P(\Phi) = P(\Phi_G) P(\Phi_S|\Phi_G) P(\theta_\Phi|\Phi_S, \Phi_G)$$

The prior over grammar formalisms  $P(\Phi_G)$  will be kept fixed at *bracketing inversion transduction grammar* in this paper. In our previous work on minimizing description length, the model length depended purely on the grammar structure, which was what we were trying to induce. Reusing that in the Bayesian interpretation gives:

$$P(\Phi_S|\Phi_G) = 2^{-DL(\Phi_S|\Phi_G)}$$

The next section contains details about how the description length of ITGs is calculated. For the parameter prior  $P(\theta_\Phi|\Phi_S, \Phi_G)$ , we choose a symmetric Dirichlet distribution over rule right-hand sides given rule left-hand sides, with a concentration parameter of two ( $\alpha_0 = \alpha_1 = \dots = \alpha_{R_i-1} = 2$  for all  $i$ ).

The full search problem we are trying to solve is thus:

$$\begin{aligned} \operatorname{argmax}_{\Phi_G, \Phi_S, \theta_\Phi} & P(\Phi_G) \times P(\Phi_S|\Phi_G) \\ & \times P(\theta_\Phi|\Phi_S, \Phi_G) \times P(D|\Phi_G, \Phi_S, \theta_\Phi) \end{aligned}$$

or conversely:

$$\begin{aligned} \operatorname{argmin}_{\Phi_G, \Phi_S, \theta_\Phi} & DL(\Phi_G) + DL(\Phi_S|\Phi_G) \\ & + DL(\theta_\Phi|\Phi_S, \Phi_G) + DL(D|\theta_\Phi, \Phi_S, \Phi_G) \end{aligned}$$

As stated earlier, we will keep  $\Phi_G$  fixed so that we are only considering bracketing inversion transduction grammars.

### 2.3 Description length of ITGs

As mentioned, the structural prior of an ITG is based on its description length. To compute the description length of an ITG, we will turn to information theory, which can be used to compute the space requirements for encoding a sequence of symbols. This requires the serialization of ITGs into sequences of symbols. To serialize an ITG, we first need to determine the alphabet that the message will be written in. We need one symbol for every nonterminal,  $L_0$ -terminal and  $L_1$ -terminal. We will also make the assumption that all these symbols are used in at least one rule, so that it is sufficient to serialize the rules in order to express the entire grammar. To serialize the rules,

we need some kind of delimiter to know where one rule ends and the next starts; we will exploit the fact that we also need to specify whether the rule is straight or inverted (unary rules are assumed to be straight), and merge these two functions into one symbol. This gives the union of the symbols of the grammar and the set  $\{\langle \rangle, \langle \rangle\}$ , where  $\langle \rangle$  signals the beginning of a straight rule, and  $\langle \rangle$  signals the beginning of an inverted rule. The serialized format of a rule will be: rule type/start marker, followed by the left-hand side nonterminal, followed by all right-hand side symbols. The symbols on the right-hand sides are either nonterminals or biterminals. The serialized form of a grammar is the serialized form of all rules concatenated.

Consider the following toy grammar:

$$\begin{aligned} S &\rightarrow A & A &\rightarrow \langle AA \rangle & A &\rightarrow [AA] \\ A &\rightarrow \text{have/有} & A &\rightarrow \text{yes/有} & A &\rightarrow \text{yes/是} \end{aligned}$$

Its serialized form would be:

$$\langle \rangle SA \langle \rangle AAA \langle \rangle AAA \langle \rangle A \text{have} \text{有} \langle \rangle A \text{yes} \text{有} \langle \rangle A \text{yes} \text{是}$$

Now we can, again turn to information theory to arrive at an encoding for this message. Assuming a uniform distribution over the symbols, each symbol will require  $-\lg \frac{1}{N}$  bits to encode (where  $N$  is the number of different symbols—the type count). The above example has 8 symbols, meaning that each symbol requires 3 bits. The entire message is 23 symbols long, which means that we need 69 bits to encode it.

### 3 Initializing model structure: Initial ITG rules

To tackle the pitfalls of premature pruning in our earlier rule-chunking approaches of starting out with a fairly general transduction grammar and fitting it to the training data (Saers *et al.*, 2011, 2012), we do the exact opposite here: we start with a transduction grammar that fits the training data as well as possible, and generalize from there. The transduction grammar that fits the training data the best is the one where the start symbol rewrites to the full sentence pairs that it has to generate. It is also possible to add any number of nonterminal symbols in the layer between the start symbol and the bisentences without altering the probability of the training data. We take advantage of this by allowing for one intermediate symbol so that the ITG conforms to the normal form and always rewrites

the start symbol to precisely one nonterminal symbol. Our initial ITG thus contains long rules that look like this:

$$\begin{aligned}
S &\rightarrow A \\
A &\rightarrow e_{0..T_0}/f_{0..V_0} \\
A &\rightarrow e_{0..T_1}/f_{0..V_1} \\
&\dots \\
A &\rightarrow e_{0..T_N}/f_{0..V_N}
\end{aligned}$$

where  $S$  is the start symbol,  $A$  is the nonterminal,  $N$  is the number of sentence pairs in the training corpus,  $T_i$  is the length of the  $i^{\text{th}}$  output sentence,  $V_i$  is the length of the  $i^{\text{th}}$  input sentence,  $e_{0..T_i}$  is the sequence  $e_{0e_1} \dots e_{T_i-1}$  of output tokens (that is: the  $i^{\text{th}}$  output sentence), and  $f_{0..V_i}$  is the sequence  $f_0f_1 \dots f_{V_i-1}$  of input tokens (that is: the  $i^{\text{th}}$  input sentence).

#### 4 Generalizing model structure: Shortening long ITG rules

To generalize the initial inversion transduction grammar we need to identify parts of the existing biterminals that could be validly used in isolation, and allow them to combine with other segments. This is the very feature that allows a *finite* transduction grammar to generate an *infinite* set of sentence pairs; doing this, moves some of the probability mass which was concentrated in the training data out to other data that are still unseen—the very notion of generalizing beyond the training data.

In practice, we will segment the existing lexical rules into smaller lexical rules and the structural rules needed to compose them into the original, unsegmented, lexical unit. This preserves the capability to generate the original transduction, but also allows for novel combinations of the newly introduced lexical building blocks into novel sentence pairs, which extends the set of sentence pairs that the grammar can generate. Although it is possible to segment one rule at a time, as we did in Saers *et al.* (2013c), it is better to collect several rules with something in common and exploit this commonality, as we did in Saers *et al.* (2013a,b). Compared to these previous works, the objective criterion that we use to drive structural generalization, as defined in Section 2.2, is also a further improvement; our shift here from an MDL to a MAP interpretation naturally suggests the enhanced formulation of the Bayesian priors.

The general strategy is to propose a number of sets of biterminal rules and a place to segment them, estimate the posterior probability given these sets and commit to the best. That is: we do a greedy search over the power set of possible segmentations of the rule set. As we will see, this intractable problem can be reasonably efficiently approximated.

The key component in the approach is the ability to evaluate the change in a posteriori probability if a specific segmentation was made in the grammar. This can then be extended to a set of segmentations, which only leaves the problem of generating suitable sets of segmentations.

In this work, we are only considering segmentation of lexical rules, which keeps the ITG in normal form, greatly simplifying processing without altering the expressivity. A lexical ITG rule has the form  $A \rightarrow e_{0..T}/f_{0..V}$ , where  $A$  is the left-hand side nonterminal—the category,  $e_{0..T}$  is a sequence of  $T$  (from position 0 up to but not including position  $T$ )  $L_0$  tokens and  $f_{0..V}$  is a sequence of  $V$  (from position 0 up to but not including position  $V$ )  $L_1$  tokens. When segmenting this rule, three new rules are produced which take one of the following forms depending on whether the segmentation is inverted or not:

$$\begin{aligned}
A &\rightarrow [BC] & A &\rightarrow \langle BC \rangle \\
B &\rightarrow e_{0..S}/f_{0..U} & \text{or} & B \rightarrow e_{0..S}/f_{U..V} \\
C &\rightarrow e_{S..T}/f_{U..V} & C &\rightarrow e_{S..T}/f_{0..U}
\end{aligned}$$

All possible splits of the terminal rule can be accounted for by choosing the identities of  $B$ ,  $C$ ,  $S$  and  $U$ , as well as whether the split is straight or inverted.

The key to a successful segmentation is to maximize the potential for reuse. Any segment that can be reused maximizes the model prior. Consider the lexical rule:

$$\begin{aligned}
A &\rightarrow \text{five thousand yen is my limit/} \\
&\quad \text{我最多出五千日元}
\end{aligned}$$

(Chinese pinyin romanization: *wǒ zuì dōu chū wǔ qiān rì yuán*). This rule can be split into three rules:

$$\begin{aligned}
A &\rightarrow \langle AA \rangle, \\
A &\rightarrow \text{five thousand yen/五千日元}, \\
A &\rightarrow \text{is my limit/我最多出}
\end{aligned}$$

Note that the original rule consists of 16 symbols (in our encoding scheme), whereas the three new

rules consists of  $4 + 9 + 9 = 22$  symbols. Add to that that three rules are likely to be less probable than one rule when parsing, which makes the training data less likely as well. It is reasonable to believe that the bracketing inverted rule  $A \rightarrow \langle AA \rangle$  is present in the grammar already, but this still leaves 18 symbols, which is decidedly longer than 16 symbols—and we need to get the length to be shorter if we want to see a net gain. What we really need to do is find a way to reuse the lexical rules that came out of the segmentation. Now suppose the grammar also contained this lexical rule:

$A \rightarrow$  the total fare is five thousand yen/  
 总共的费用是五千日元

(Chinese pinyin romanization: *zōng gòng de fèi yòng shì wǔ qiān rì yuán*). This rule can also be split into three rules:

$A \rightarrow [AA]$ ,  
 $A \rightarrow$  the total fare is/总共的费用是,  
 $A \rightarrow$  five thousand yen/五千日元

Again, we will assume that the structural rule is already present in the grammar, the old rule was 19 symbols long, and the two new terminal rules are  $12 + 9 = 21$  symbols long. Again we are out of luck, as the new rules are longer than the old one. The way to make this work is to realize that the two existing rules share a bilingual affix—a **bi-affix**: five thousand dollars translating into 五千日元. If we make the two changes at the same time, we get rid of  $16 + 19 = 35$  symbols worth of rules, and introduce a mere  $9 + 9 + 12 = 30$  symbols worth of rules (assuming the structural rules are already in the grammar). Making these two changes at the same time is essential, as the length of the five saved symbols can be used to offset the likely decrease in the probability of the data given the grammar. And of course: the more rules we can find with shared biaffixes, the more likely we are to find a good set of segmentations.

Our algorithm takes advantage of the above observation by focusing on the biaffixes found in the training data. Each biaffix defines a set of lexical rules paired up with a possible segmentation. We evaluate the biaffixes by estimating the change in posterior probability associated with committing to all the segmentations defined by a biaffix. This allows us to find the best set of segmentations, but rather than committing only to the one best set of

---

**Algorithm 1** Bayesian learning of ITG structure through iterative rule segmentation.

---

```

 $\Phi$                                 ▷ The ITG being induced
repeat
   $\delta \leftarrow 1$ 
   $bs \leftarrow \text{collect\_biaffixes}(\Phi)$ 
   $b\delta \leftarrow []$ 
  for all  $b \in bs$  do
     $\delta_b \leftarrow \text{eval\_map}(b, \Phi)$ 
    if  $\delta_b > 1$  then
       $b\delta \leftarrow [b\delta, \langle b, \delta_b \rangle]$ 
    end if
  end for
   $\text{sort\_by\_delta}(b\delta)$ 
  for all  $\langle b, \delta_b \rangle \in b\delta$  do
     $\delta'_b \leftarrow \text{eval\_map}(b, \Phi)$ 
    if  $\delta'_b > 1$  then
       $\Phi \leftarrow \text{make\_segmentations}(b, \Phi)$ 
       $\delta \leftarrow \delta \delta'_b$ 
    end if
  end for
until  $\delta \leq 1$ 
return  $\Phi$ 

```

---

segmentations, we will collect all sets which would improve the posterior probability, and try to commit to as many of them as possible. This minimizes the parsing efforts, which are very expensive.

The pseudocode for the search algorithm can be found in Algorithm 1. It uses the methods `collect_biaffixes`, `eval_map`, `sort_by_delta` and `make_segmentations`, which collects all biaffixes found in the rules of an ITG, evaluates the change in posterior probability caused by segmenting an ITG according to a biaffix, sorts biaffix–change-in-posterior pairs according to the latter, and commits to a set of segmentations, respectively.

To evaluate the change in posterior probability caused by a proposed set of candidate segmentations, we need to calculate the ratio between the posterior of the current model structure and the model structure that would result from committing to the candidate segmentations:

$$\frac{P(\Phi'|D)}{P(\Phi|D)} \propto \frac{P(\Phi'_S|\Phi_G) P(D|\Phi'_S, \Phi_G, \theta_{\Phi'})}{P(\Phi_S|\Phi_G) P(D|\Phi_S, \Phi_G, \theta_{\Phi})}$$

The proportionality holds because we are keeping the model formalism ( $\Phi_G$ ) and the model parameters ( $\theta_{\Phi}$  and  $\theta_{\Phi'}$ ) fixed. We arrive at the ratio between the structural priors by using description



length:

$$\frac{P(\Phi'_S|\Phi_G)}{P(\Phi_S|\Phi_G)} = 2^{-(\text{DL}(\Phi'_S) - \text{DL}(\Phi_S))}$$

Rather than biparsing the entire training data with the two models, we approximate the change as the ratio between the probabilities of the rules that differ. We thus assume that:

$$\frac{P(D|\Phi'_S, \Phi_G, \theta_{\Phi'})}{P(D|\Phi_S, \Phi_G, \theta_{\Phi})} = \frac{\hat{p}'(r_1)\hat{p}'(r_2)\hat{p}'(r_3)}{\hat{p}(r_0)}$$

where  $\hat{p}$  and  $\hat{p}'$  are the estimated rule probability functions within  $\theta_{\Phi}$  and  $\theta_{\Phi'}$  respectively. They differ only with respect to the changed rules, such that:

$$\begin{aligned}\hat{p}'(r_0) &= 0 \\ \hat{p}'(r_1) &= \hat{p}(r_1) + \frac{1}{3}\hat{p}(r_0) \\ \hat{p}'(r_2) &= \hat{p}(r_2) + \frac{1}{3}\hat{p}(r_0) \\ \hat{p}'(r_3) &= \hat{p}(r_3) + \frac{1}{3}\hat{p}(r_0)\end{aligned}$$

When more than one rule is segmented, we first aggregate the changes in rule probabilities for the entire set of rules, and then aggregate the changes in data probability.

We have now approximated the change in posterior probability to the point that we can efficiently calculate it in closed form for an arbitrary set of rule segmentations.

For practical purposes, we perform the search in two phases: one that focuses on the structure of the ITG, and one that focuses on the probabilities. The former performs top-down rule segmentation as described in Saers *et al.* (2013b), adjusting  $\Phi_S$  to optimize the posterior (thus affecting the prior over the structure of the ITG  $P(\Phi_S|\Phi_G)$  and the conditional probability of the data given the complete model  $P(D|\Phi_G, \Phi_S, \theta_{\Phi})$ ). The latter adjusts the model parameters  $\theta_{\Phi}$  to optimize the posterior (thus affecting the prior over the parameters  $P(\theta_{\Phi}|\Phi_S, \Phi_G)$  and again  $P(D|\Phi_G, \Phi_S, \theta_{\Phi})$ ), assuming the model structure  $\Phi_S$  to be fixed, as well as  $\Phi_G$  which remains fixed as bracketing inversion transduction grammars.

The prior is a symmetric Dirichlet distribution over rule right-hand sides given rule left-hand sides. To get the conditional, we have to biparse the training data, and to maximize it, we perform expectation maximization (Dempster *et al.*,

1977), as specified for ITGs by (Wu, 1995) with the caveat that we increase all the fractional counts by one before normalizing. The biparsing is done with our in-house implementation of the cubic time biparsing algorithm described in Saers *et al.* (2009), with a beam width of 100.

## 5 Experimental setup

To test the viability of the idea of starting with a very specific ITG consisting of long rules, and iteratively segmenting the rules to induce a more general ITG under a MAP or MDL objective, we have implemented the steps detailed in Sections 3 and 4; in this section we will describe in greater detail the exact experimental conditions of our empirical study.

The initial BITG is set to have the relative frequency of the unique sentences as the probability of the corresponding rules. This parametrization is identical to what we would have arrived at with any other initialization that was subsequently optimized with expectation maximization; in this case it is possible to jump straight to the optimum. The generalization step requires biparsing in order to estimate the posterior—we use the cubic time biparsing algorithm described in Saers *et al.* (2009), with a beam width of 100. In the parameter optimization step, we use the exact same biparser.

As training data, we use the IWSLT07 Chinese–English data set (Fordyce, 2007), which contains 46,867 sentence pairs of training data, and 489 Chinese sentences with 6 English reference translations each as test data; all the sentences are taken from the traveling domain. Since the Chinese is written without whitespace, we use a tool which tries to clump characters together into more “word-like” sequences (Wu, 1999).

After each induction iteration there is a fully-functional grammar that we can test as a translation system. For this, we use our in-house ITG decoder, which uses a CKY-style parsing algorithm (Cocke, 1969; Kasami, 1965; Younger, 1967) with cube pruning (Chiang, 2007) to integrate the language model scores. We use SRILM (Stolcke, 2002) to train a trigram language model on the English side of the training data.

To evaluate the resulting translations, we use BLEU (Papineni *et al.*, 2002), and NIST (Dodington, 2002), and compare the results against our bottom-up oriented chunking ITG induction approach (Saers *et al.*, 2012).

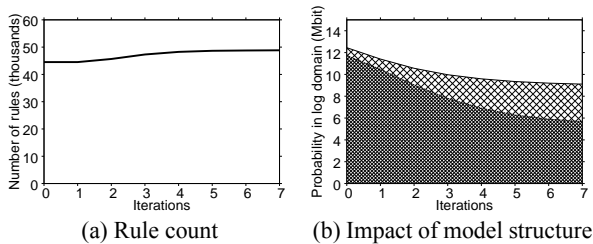


Figure 1: Number of rules (a), and the impact of changes in the model structure (b) during the structure induction phase. The change in model structure is broken down into the model prior (bottom) and data given model (top).

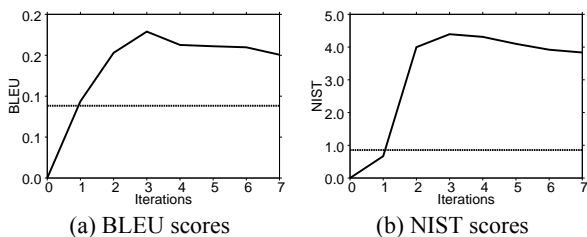


Figure 2: Variations in translation quality over different iterations. The dotted line represents the baseline (Saers *et al.*, 2012).

## 6 Results

We need to evaluate (a) how well the introduced induction works, and (b) how well the resulting model works. How well the induction works can be seen in Figure 1, which shows how the model changes over iterations. Although the number of rules rises, the model structure prior becomes more probable, indicating that smaller rules are being learned. The improvements in the prior fully makes up for the loss in the probability of the data given the model, which indicates that we are indeed generalizing successfully. The translation quality of the resulting model is found in Table 1 and Figure 2, which show how the translation quality changes as measured by two automatic quality metrics (Papineni *et al.*, 2002; Doddington, 2002). It is clear that the maximum *a posteriori* probability objective pushes the top-down learning approach far past the maximum likelihood objective of the bottom-up chunking learning approach, which is the baseline we are comparing against.

## 7 Conclusions

We have introduced a minimalist model for unsupervised Bayesian induction of parsimonious

Table 1: Translation results of the baseline, the initial model and the model after  $n$  iterations.

System	NIST	BLEU
baseline	0.8554	8.83
initial	0.0000	0.00
iteration 1	0.6686	9.38
iteration 2	3.9976	15.30
iteration 3	<b>4.3928</b>	<b>17.89</b>
iteration 4	4.3122	16.26
iteration 5	4.0981	16.10
iteration 6	3.9191	15.97
iteration 7	3.8338	15.06

phrasal ITGs, and shown that iteratively splitting existing rules into smaller rules driven by a maximum a posteriori probability objective is superior to iteratively chunking atomic rules into longer rules driven by a maximum likelihood objective. A novel top-down segmenting search strategy allows for efficient prediction of changes in posterior probability of the data given changes in the model—a key ingredient for MAP training. Decoding is done directly with induced transduction grammars, a more “pure” evaluation methodology than embedding them within many other heuristic components that obscure induction characteristics. This provides an obvious foundation for generalization to more general transduction grammars.

## Acknowledgements

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

## References

- Phil BLUNSOM and Trevor COHN. “Inducing synchronous grammars with slice sampling.” *NAACL HLT 2010*, 238–241. Los Angeles, CA, Jun 2010.
- Phil BLUNSOM, Trevor COHN, Chris DYER, and Miles OSBORNE. “A gibbs sampler for phrasal synchronous grammar induction.” *ACL-IJCNLP 2009*, 782–790. Suntec, Singapore, Aug 2009.

- Phil BLUNSON, Trevor COHN, and Miles OSBORNE. “Bayesian synchronous grammar induction.” *NIPS 21*. Vancouver, Canada, Dec 2008.
- David BURKETT, John BLITZER, and Dan KLEIN. “Joint parsing and alignment with weakly synchronized grammars.” *NAACL HLT 2010*, 127–135. Los Angeles, CA, Jun 2010.
- Stanley F. CHEN. “Bayesian grammar induction for language modeling.” *ACL 95*, 228–235. Cambridge, MA, Jun 1995.
- Colin CHERRY and Dekang LIN. “Inversion transduction grammar for joint phrasal translation modeling.” *SSST*, 17–24. Rochester, NY, Apr 2007.
- David CHIANG. “A hierarchical phrase-based model for statistical machine translation.” *ACL-05*, 263–270. Ann Arbor, MI, Jun 2005.
- David CHIANG. “Hierarchical phrase-based translation.” *Computational Linguistics*, 33(2):201–228, 2007.
- John COCKE. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.
- Arthur Pentland DEMPSTER, Nan M. LAIRD, and Donald Bruce RUBIN. “Maximum likelihood from incomplete data via the em algorithm.” *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- George DODDINGTON. “Automatic evaluation of machine translation quality using n-gram co-occurrence statistics.” *HLT '02*, 138–145. San Diego, CA, 2002.
- C. S. FORDYCE. “Overview of the IWSLT 2007 evaluation campaign.” *IWSLT 2007*, 1–12. 2007.
- Michel GALLEY, Jonathan GRAEHL, Kevin KNIGHT, Daniel MARCU, Steve DENEEFE, Wei WANG, and Ignacio THAYER. “Scalable inference and training of context-rich syntactic translation models.” *COLING/ACL 2006*, 961–968. Sydney, Australia, Jul 2006.
- Aria HAGHIGHI, John BLITZER, John DENERO, and Dan KLEIN. “Better word alignments with supervised itg models.” *ACL-IJCNLP 2009*, 923–931. Suntec, Singapore, Aug 2009.
- Howard JOHNSON, Joel MARTIN, George FOSTER, and Roland KUHN. “Improving translation quality by discarding most of the phrasetable.” *EMNLP-CoNLL 2007*, 967–975. Prague, Czech Republic, Jun 2007.
- Tadao KASAMI. “An efficient recognition and syntax analysis algorithm for context-free languages.” *Tech. Rep. AFCRL-65-00143*, Air Force Cambridge Research Laboratory, 1965.
- Philipp KOEHN, Franz Joseph OCH, and Daniel MARCU. “Statistical Phrase-Based Translation.” *HLT-NAACL 2003*, vol. 1, 48–54. Edmonton, Canada, May/June 2003.
- Graham NEUBIG, Taro WATANABE, Shinsuke MORI, and Tatsuya KAWAHARA. “Machine translation without words through substring alignment.” *ACL 2012*, 165–174. Jeju Island, Korea, Jul 2012.
- Graham NEUBIG, Taro WATANABE, Eiichiro SUMITA, Shinsuke MORI, and Tatsuya KAWAHARA. “An unsupervised model for joint phrase alignment and extraction.” *ACL HLT 2011*, 632–641. Portland, OR, Jun 2011.
- Kishore PAPINENI, Salim ROUKOS, Todd WARD, and Wei-Jing ZHU. “BLEU: a method for automatic evaluation of machine translation.” *ACL-02*, 311–318. Philadelphia, PA, Jul 2002.
- Jason RIESA and Daniel MARCU. “Hierarchical search for word alignment.” *ACL 2010*, 157–166. Uppsala, Sweden, Jul 2010.
- Jorma RISSANEN. “A universal prior for integers and estimation by minimum description length.” *The Annals of Statistics*, 11(2):416–431, Jun 1983.
- Markus SAERS, Karteek ADDANKI, and Dekai WU. “From finite-state to inversion transductions: Toward unsupervised bilingual grammar induction.” *COLING 2012*, 2325–2340. Mumbai, India, Dec 2012.
- Markus SAERS, Karteek ADDANKI, and Dekai WU. “Combining top-down and bottom-up search for unsupervised induction of transduction grammars.” *SSST-7*, 48–57. Atlanta, GA, Jun 2013a.
- Markus SAERS, Karteek ADDANKI, and Dekai WU. “Iterative rule segmentation under minimum description length for unsupervised transduction grammar induction.” Adrian-Horia DEDIU, Carlos MARTÍN-VIDE, Ruslan MITKOV, and Bianca TRUTHE (eds.), *Statistical Language and Speech Processing, First International Conference, SLSP 2013*, Lecture Notes in Artificial Intelligence (LNAI). Tarragona, Spain: Springer, Jul 2013b.
- Markus SAERS, Karteek ADDANKI, and Dekai WU. “Unsupervised transduction grammar induction via minimum description length.” *HyTra*, 67–73. Sofia, Bulgaria, Aug 2013c.
- Markus SAERS, Joakim NIVRE, and Dekai WU. “Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm.” *IWPT'09*, 29–32. Paris, France, Oct 2009.
- Markus SAERS, Joakim NIVRE, and Dekai WU. “Word alignment with stochastic bracketing linear inversion transduction grammar.” *NAACL HLT 2010*, 341–344. Los Angeles, CA, Jun 2010.
- Markus SAERS and Dekai WU. “Improving phrase-based translation via word alignments from Stochastic Inversion Transduction Grammars.” *SSST-3*, 28–36. Boulder, CO, Jun 2009.
- Markus SAERS and Dekai WU. “Principled induction of phrasal bilexica.” *EMT-2011*, 313–320. Leuven, Belgium, May 2011.
- Markus SAERS, Dekai WU, Chi KIU LO, and Karteek ADDANKI. “Speech translation with grammar driven probabilistic phrasal bilexica extraction.” *Interspeech 2011*, 2089–2092. 2011.
- Claude Elwood SHANNON. “A mathematical theory of communication.” *The Bell System Technical Journal*, 27:379–423, 623–656, Jul, Oct 1948.
- Ray J. SOLOMONOFF. “A new method for discovering the grammars of phrase structure languages.” *IFIP*, 285–289. 1959.
- Andreas STOLCKE. “SRILM – an extensible language modeling toolkit.” *ICSLP2002 - INTERSPEECH 2002*, 901–904. Denver, CO, Sep 2002.
- Andreas STOLCKE and Stephen OMOHUNDRO. “Inducing probabilistic grammars by bayesian model merging.” R. C. CARRASCO and J. ONCINA (eds.), *ICGI-94*, 106–118. Springer, 1994.
- Juan Miguel VILAR. “Experiments using mar for aligning corpora.” *ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, 95–98. Ann Arbor, Jun 2005.
- Juan Miguel VILAR and Enrique VIDAL. “A recursive statistical translation model.” *ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, 199–207. Ann Arbor, Jun 2005.
- Dekai WU. “Trainable coarse bilingual grammars for parallel text bracketing.” *WVLC-3*, 69–81. Cambridge, MA, Jun 1995.
- Dekai WU. “Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora.” *Computational Linguistics*, 23(3):377–403, 1997.
- Zhibiao WU. “LDC Chinese segmenter.” 1999.
- Daniel H. YOUNGER. “Recognition and parsing of context-free languages in time  $n^3$ .” *Information and Control*, 10(2):189–208, 1967.
- Hao ZHANG, Chris QUIRK, Robert C. MOORE, and Daniel GILDEA. “Bayesian learning of non-compositional phrases with synchronous parsing.” *ACL-08: HLT*, 97–105. Columbus, OH, Jun 2008.

# Estimating the Quality of Translated User-Generated Content

Raphael Rubino<sup>†‡</sup>, Jennifer Foster<sup>†</sup>, Rasoul Samad Zadeh Kaljahi<sup>†‡</sup>,  
Johann Roturier<sup>‡</sup> and Fred Hollowood<sup>‡</sup>

<sup>†</sup>NCLT, School of Computing, Dublin City University, Ireland  
{rrubino, jfoster, rkaljahi}@computing.dcu.ie

<sup>‡</sup>Symantec Research Labs, Dublin, Ireland  
{johann\_roturier, fhollowood}@symantec.com

## Abstract

Previous research on quality estimation for machine translation has demonstrated the possibility of predicting the translation quality of well-formed data. We present a first study on estimating the translation quality of *user-generated content*. Our dataset contains English technical forum comments which were translated into French by three automatic systems. These translations were rated in terms of both comprehensibility and fidelity by human annotators. Our experiments show that tried-and-tested quality estimation features work well on this type of data but that extending this set can be beneficial. We also show that the performance of particular types of features depends on the type of system used to produce the translation.

## 1 Introduction

Quality Estimation (QE) involves judging the correctness of a system output given an input without any output reference. Substantial progress has been made on QE for Machine Translation (MT), but research has been mainly conducted on well-formed, edited text (Blatz et al., 2003; Ueffing et al., 2003; Raybaud et al., 2009; Specia et al., 2009). We turn our attention to estimating the quality of *user-generated content* (UGC) translation – a particularly relevant use of QE since the translation process is likely to be affected by the noisy nature of the input, particularly if the MT system is trained on well-formed text.

The source language content is collected from an IT Web forum in English and translated into French by three automatic systems. For each MT system, the produced translation is manually evaluated following two criteria: the translation

comprehensibility and fidelity. We evaluate several feature sets on the UGC dataset including the baseline suggested by the organisers of the WMT 2012 QE for MT shared task (Callison-Burch et al., 2012) and a feature set designed to model typical characteristics of forum text.

The novel contributions of the paper are: 1) testing the WMT QE for MT Shared Task baseline feature set on the UGC dataset and demonstrating its portability, 2) introducing new features which contribute to significant performance gains on both QE tasks, and 3) building three different QE systems using three different MT systems and showing that the usefulness of a feature type depends on the MT system although better performance can be achieved by training a QE system on the combined output of the three systems.

The paper is organised as follows. Related work on QE for MT is described in Section 2, followed in Section 3 by a description of the dataset. We describe the QE features in Section 4 and present the results of our experiments in Section 5. A discussion of the results, as well as a comparison with previous work, are presented in Section 6. Finally, we conclude and suggest future work in Section 7.

## 2 Background

The main approach for QE in MT is based on estimating how correct MT output is through characteristic elements extracted from the source and the target texts and the MT system involved in the translation process. These elements, or features, are seen as predictive parameters that can be combined with machine learning methods to estimate binary, multi-class, or continuous scores. First applied at the word level (Gandraber and Foster, 2003; Ueffing et al., 2003), QE for MT was then extended to the sentence level during a workshop in the same year (Blatz et al., 2003).

Many different feature sources have been used including surface features (segment length, punc-

tuation marks, etc.), language model features (perplexity, log-probability, etc.), word or phrase alignment features,  $n$ -best list features, internal MT system scores (Quirk, 2004; Ueffing and Ney, 2004), and linguistic features (Gamon et al., 2005; Specia and Gimenez, 2010). In a recent study, features based on the intra-language mutual information between words and backward language models were introduced (Raybaud et al., 2011). Other studies evaluate the gain brought by features extracted from MT output back-translation (Albrecht and Hwa, 2007), pseudo-references in the form of output from other MT systems for the same source sentence (Soricut et al., 2012), and topic models (Rubino et al., 2012).

Previous studies also differ on the labels to predict: binary scores (Quirk, 2004) or continuous scores such as those given by automatic metrics (Bojar et al., 2013) or averaged human evaluations (Specia et al., 2009; Callison-Burch et al., 2012). As regards the learning algorithms used, several have been tried, with support vector machine and decision tree learning proving popular (Callison-Burch et al., 2012).

### 3 Dataset

We use the dataset presented in Roturier and Bensadoun (2011), which was obtained by machine-translating 694 English segments, harvested from the Symantec English Norton forum<sup>1</sup>, into French using three different translators (MOSES (Koehn et al., 2007), MICROSOFT<sup>2</sup> (MS) and SYSTRAN). The translations were then evaluated in terms of comprehensibility (1 to 5 scale) and fidelity (binary scale) by human annotators. The source side of this data set represents user-generated content – see Banerjee et al. (2012) for a detailed description of the characteristics of this type of data and see Table 2 for some examples. For each of the three translators, we extract 500 segments from this dataset to build our training sets. The remaining 194 segments per translator are used as test sets. The distribution of the comprehensibility and fidelity classes over the three MT systems are shown in Table 1.

### 4 Quality Estimation Features

In this section, we describe the features which we added to the 17 baseline features provided by the

<sup>1</sup><http://community.norton.com>

<sup>2</sup><http://www.bing.com/translator/>

Class	Comprehensibility					Fidelity
	1	2	3	4	5	1
MOSES	6.1	55.0	12.7	11.2	15.0	37.2
MS	10.7	39.8	19.2	13.5	16.9	46.0
SYSTRAN	11.5	45.7	14.8	11.7	16.3	41.2

Table 1: Distribution (%) over the comprehensibility and fidelity classes for the 694 segments per MT system.

WMT12 QE shared task organisers to make our “extended” feature set. We then introduce a set of 37 features which relate specifically to the user-generated-content aspect of our data.

#### 4.1 Extended Feature Set

- **15 Surface Features** Average target word length, average source word occurrence, number of uppercased letters and the ratio of all source and target surface features.

- **180 Language Model Features** Source and target  $n$ -gram ( $n \in [1; 5]$ ) log-probabilities and perplexities on two LMs built on the seventh version of Europarl and the eighth version of News-Commentary (30 features). The same number of features are extracted from a backward version of these two LMs (Duchateau et al., 2002). We repeat this feature extraction process using four LMs built on the Symantec Translation Memories (TMs)<sup>3</sup> and four LMs built on the monolingual Symantec forum data<sup>4</sup>.

- **15 MT Output Language Model Features** A MOSES English-French PB-SMT system is trained on the Symantec TMs and the same target LM used to extract the baseline LM features. The English side of the Symantec Norton monolingual forum data is translated by this system and the output is used to build a 5-gram LM. Target features are then extracted in a similar way as the standard LM features.

- **4 Word Alignment Features** Using GIZA++ (Och and Ney, 2000) and the Symantec TMs, word alignment probabilities are extracted from the source and target segments.

- **78  $n$ -gram Frequency Features** The number of source and target segments unigrams seen in a reference corpus plus the percentage of  $n$ -grams in frequency quartiles ( $n \in [1; 5]$ ). The reference corpus is the same corpus used to extract the LM features.

<sup>3</sup>  $\sim 1.6M$  aligned segments in English and French.

<sup>4</sup>  $\sim 3M$  segments in English and  $\sim 40k$  segments in French.

---

*so loe and behold I get a Internet Worm Protection Signature File Version: 20090511.001. on 5/20/09 in the afternoon*  
*Start NIS 2009 > In the Internet pane, click Settings > Under Smart Firewall, click configure next to Advanced settings >*  
*In the Advanced Settings window, turn off Automatic Printer Sharing control.*

---

*ok then what should do am i safe as is meaning just leave it alone as long it get blocked it can get my info right i have*  
*no clew how get rid of it the only thing i could do that i know would work is to take everthing out my computer and format*  
*it with boo disk will this work?*

---

Table 2: Processing challenges associated with forum text: some examples.

- **9 Back-translation Features** We translate the target segments back into the source language using 3 different MT systems: MS, SYSTRAN and a MOSES PB-SMT system (trained on the Symantec TMs) and we measure the distance between the original source segments and the back-translated ones using BLEU, TER and Levenshtein.

- **23 Topic Features** Following Rubino et al. (2012; 2013), a bilingual topic model based on Latent Dirichlet Allocation (Blei et al., 2003) is built using the Symantec TMs. 20 features are the source and target segment distributions over the 10-dimensional topic space and 3 features are the distances between these distributions, using the cosine, euclidean distance and city-block metrics.

- **16 Pseudo-reference Features** Following Soricut et al. (2012), we compare each MT system output to the two others using sentence-level BLEU, error information provided by TER (no. of insertions, deletions, etc.) and Levenshtein.

- **3 Part-of-Speech Features** We count the number of POS tag types in the source and target segments, extracted from trees produced by the Stanford parser (Klein and Manning, 2003). The ratio of these two values is also included.

#### 4.2 UGC-related Features

We also experiment with features that capture the noisy nature of UGC. Some are related to the inconsistent use of character case, some to non-standard punctuation, some to spelling mistakes and some to the tendency of sentence splitters to underperform on this type of text. From each source-target pair, we extract the following information (in the form of one feature for the source segment, one for the target segment and, where appropriate, one for the ratio between the two):

- **11 Case Features** the number of upper and lowercased words, the number of fully uppercased words, the number of mixed-case words and whether or not the segment begins with an upper-case letter.

- **13 Punctuation Features** the ratio between punctuation characters and other characters, the

number of words containing a full stop, the number of sentences produced by an off-the-shelf sentence splitter for each segment (included in NLTK (Bird, 2006)), whether or not the segment contains a dash, an ellipsis, and whether or not the segment ends with a punctuation symbol.

- **9 Acronym and Emotion Features** the number of web and IT-domain acronyms and the number of emoticons.

- **4 Linguistic Features** the number of spelling mistakes flagged by the spellchecker LANGUAGE-TOOL<sup>5</sup> and whether or not the segment starts with a verb (indicating imperatives or questions).

## 5 Experiments

Classification models are built using the  $C$ -SVC implementation in LIBSVM (Chang and Lin, 2011) with a Radial Basis Function (RBF) kernel. Optimal hyper-parameters  $C$  and  $\gamma$  are found by grid-search with a 5-fold cross-validation on the training set (optimising for accuracy). For evaluation, we measure the accuracy, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). All the results are compared to the baseline for significance testing using bootstrap resampling. We present the results on the comprehensibility task first, followed by the results on the fidelity task. In order to remove noisy and redundant features we also experiment with feature selection. We try several approaches<sup>6</sup> and report results with the approach that performs best during cross-validation on the training set.

### 5.1 Translation Comprehensibility Results

The full set of comprehensibility estimation results are presented in Table 3. We see that a higher classification accuracy does not necessarily imply lower MAE and RMSE, e.g. the MOSES and

<sup>5</sup><http://www.language-tool.org/>

<sup>6</sup>These include information gain univariate filtering, correlation-based multivariate filtering, a naive Bayes wrapper approach and principal component analysis. All are implemented in the WEKA machine learning toolkit (Hall et al., 2009). Of the approaches tried, none stood out as clearly superior to the others and the choice seems to depend on the task and the MT system.

SYSTRAN experiments show that the extended set leads to a higher classification accuracy compared to the baseline, while the two error scores are lower on the baseline compared to the extended set. This discrepancy can happen because the accuracy measure is not sensitive to differences between comprehensibility scores whereas the error measures are – the error measures will prefer a system which gives a 5-scoring translation a score of 4 than one which gives it a score of 1. Statistical significance tests show that the extended feature set outperforms the baseline significantly only for the system trained on MS translations. The UGC feature set seems to add useful information only for the system trained on MOSES translations.

	MOSES	MS	SYSTRAN
<i>Baseline</i>			
Acc. (%)	67.0	39.7	55.2
MAE	0.48	0.96	0.64
RMSE	<b>0.94</b>	1.38	1.07
<i>Extended</i>			
Acc. (%)	69.1	42.8*	55.7
MAE	0.51	0.87	0.65
RMSE	1.01	1.28	1.11
<i>Extended+UGC</i>			
Acc. (%)	69.1	41.8	55.2
MAE	0.49	0.89	0.67
RMSE	0.99	1.29	1.14
<i>Extended + Feature Selection</i>			
Acc. (%)	<b>70.6*</b>	39.2	<b>56.7</b>
MAE	<b>0.47</b>	0.92	<b>0.56*</b>
RMSE	0.97	1.32	<b>0.96*</b>
<i>Feature Types + Feature Selection</i>			
Acc. (%)	69.6	42.8*	52.6
MAE	0.49	0.85*	0.70
RMSE	0.99	<b>1.24*</b>	1.14
<i>Mixed-Translator: Extended+UGC</i>			
Acc. (%)	69.1	<b>44.3*</b>	54.6
MAE	0.48	<b>0.84*</b>	0.65
RMSE	0.98	1.27	1.09

Table 3: Translation comprehensibility estimation. Best results are in bold, statistically significant improvements over the baseline ( $p < 0.05$ ) are indicated with \*.

To evaluate the impact of different types of features, we conduct an evaluation by feature subset (see Figure 1). The results show that the best-performing features vary across the MT systems. The pseudo-reference and POS features are particularly useful for the system trained on MOSES translations. For the system trained on MS translations, the  $n$ -gram frequency features based on the Symantec TMs are clearly outperforming all the other feature types, with an accuracy of 42.8%. For the system trained on SYSTRAN translations, the pseudo-reference features yield an accuracy

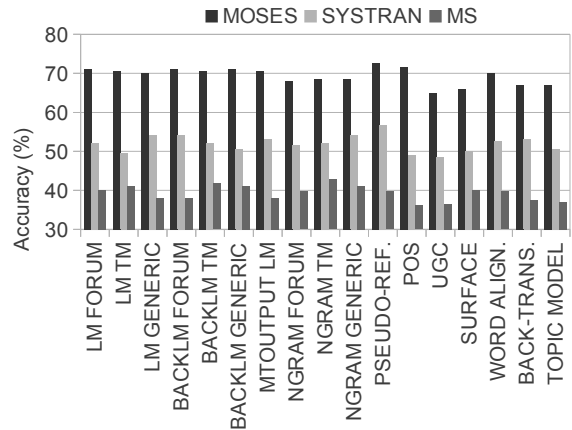


Figure 1: Feature types for comprehensibility.

score of 56.7%, which outperforms the extended set. The system trained on MS translations does not benefit from the pseudo-reference features as much as the two other QE systems. Perhaps this is because these features provide a reliable indication of translation quality when the two MT systems being compared are trained on similar data – MOSES was trained using the domain- and genre-specific data and SYSTRAN is optimized using a domain-specific lexicon, which increases the proximity of the translations generated by these two systems. LM features appear to be very useful for the three MT systems: the backward LM built on the TMs leads to the best accuracy results amongst the LM-based features for MS, while SYSTRAN and MOSES benefit from features extracted using a backward LM built on forum data.

According to the results in Fig. 1, several feature types individually outperform the baseline and the extended sets, which indicates that unsuited features are included in these two sets and motivates the application of feature selection. The feature selection algorithms are applied in two ways: on the extended set and on each feature type individually. For this second approach, the reduced feature types are combined to form the final set. The results obtained with the first and second feature selection methods are presented in the fourth and fifth rows of Table 3 respectively. The systems trained on MS and SYSTRAN translations clearly benefit from the feature selection process with significant improvement over the baseline. For the system trained on the MOSES translations, only the accuracy scores are improved over the baseline. The choice of which of the two methods of applying feature selection to use also depends on MT system.

As the training set for each MT system is small (500 instances), we combine these training sets and build a mixed-translator classification model (last row in Table 3). Note that this means that each training source segment will appear three times (one for each of the MT systems). We use the *Extended+UGC* feature set to build the mixed-translator model. Comparing to the results in the third row (individual MT classification model), we observe that it is generally beneficial to combine the translations into one larger model.

## 5.2 Translation Fidelity Results

The fidelity results are presented in Table 4.<sup>7</sup>

	MOSES	MS	SYSTRAN
Baseline	81.4	62.9	68.0
Extended	81.4	63.4	<b>76.8</b>
Extended+UGC	80.4	65.5	73.7
Extended+sel.	77.3	64.4	72.2
Type+sel.	<b>82.0</b>	<b>69.1</b>	74.2
Mixed-Translator: Ext+UGC	<b>82.0</b>	66.5	76.3

Table 4: Accuracy for fidelity estimation, best results are in bold.

According to the results for the extended feature set, the baseline result for the system trained on MOSES translations appears to be very difficult to improve upon. For this system, adding the UGC features actually degraded the accuracy scores, while it helps the system trained on MS translations. The extended set reaches the best accuracy scores (76.8%) for the system trained on SYSTRAN translations with a 8.8pt absolute improvement over the baseline set. However, statistical significance testing show that none of the improvements over the baseline are statistically significant.

As with the comprehensibility results, the impact of the feature sets depends on the MT system (see Figure 2). Again, the pseudo-reference features lead to the highest accuracy score for the system trained on MOSES translations (+2.1pts absolute compared to the baseline and extended sets) and the system trained on SYSTRAN translations (equal to the extended set), while it is not the case for the system trained on MS translations. For this latter system, the  $n$ -gram frequency features based on the forum data reach 66.0% accuracy. The accuracy results per feature type show a larger divergence compared to the results obtained on the

<sup>7</sup>We do not measure the two error scores for the fidelity scores prediction because it is a binary classification task.

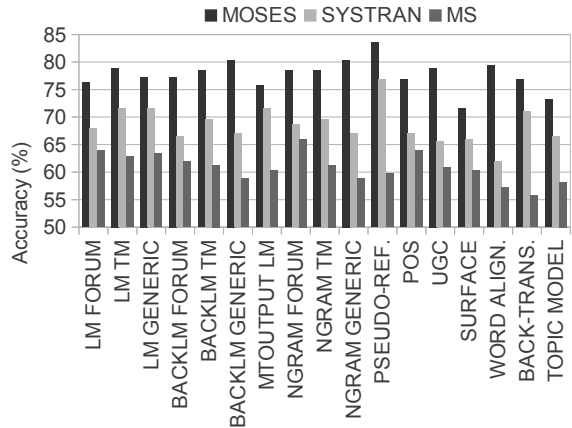


Figure 2: Feature types for fidelity.

comprehensibility task. Some feature types appear to be particularly noisy for the fidelity task, for instance the surface features for MOSES, the back-translation for MS and the word-alignment for SYSTRAN. To tackle this issue, the same feature selection methods previously used for the comprehensibility task are applied. The fourth and fifth rows in Table 4 show the results for the two methods of applying feature selection. We can see that selecting features within individual feature types leads to better results compared to applying feature selection to the full set.

As with the comprehensibility task, we build a mixed-translator fidelity estimator using the *Extended+UGC* feature set (last row in Table 4) and we observe here also that it is beneficial to combine the training data compared to training individual models.

## 6 Analysis

Adding features to the baseline set does not necessarily lead to better QE. Perhaps the baseline feature set is already diverse enough (surface, LM, word alignment, etc.). However, an error analysis shows that including the UGC features does bring useful information, especially when the source segments contain URLs, as shown in Table 5. In the case of untranslated elements, the spellchecker sometimes provides important information to the classifier about the MT output quality.

When we compare the QE results over the three MT systems, there is substantial variation. One possible explanation for this variation is the class distributions for the three sets of translations. The set whose quality is hardest to predict (MS) is the one with a more balanced distribution over the classes for both tasks (see Table 1). As classifiers



<i>Translator: MS</i>	
<b>Source</b>	How to remove status bar indicator? <URL>
<b>Target</b>	Comment supprimer l'indicateur de la barre de statut ? <URL>
	<b>Baseline</b> → 2 <b>+UGC</b> → 4 <b>Ref</b> → 4
<b>Source</b>	Best Regards Anders
<b>Target</b>	Best Regards Anders
	<b>Extended</b> → 5 <b>+UGC</b> → 1 <b>Ref</b> → 1
<i>Translator: SYSTRAN</i>	
<b>Source</b>	If you look at the URL of a Norton Safe Search results page, it contains 'search-results.com'.
<b>Target</b>	Si vous regardez l'URL d'une page de résultats de Recherche sécurisée Norton, elle contient 'search-results.com'.
	<b>Baseline</b> → 2 <b>+UGC</b> → 3 <b>Ref</b> → 3
<b>Source</b>	cgoldman wrote:
<b>Target</b>	le cgoldman s'est enregistré :
	<b>Extended</b> → 5 <b>+UGC</b> → 2 <b>Ref</b> → 2

Table 5: Example of segments where the correct comprehensibility class is predicted using UGC features.

can be biased towards the majority class, the QE task appears to be more difficult with a balanced dataset with a high standard deviation.

The best-performing feature type varies amongst the MT systems. For instance, the features built on the domain-specific translation memories (LMs and  $n$ -gram counts) bring more useful information when estimating the translation comprehensibility of the MS translations. It is possible that this is happening because the MOSES and SYSTRAN systems were trained on domain-specific data while the MS system was not. Domain-specific features may be particularly helpful in estimating the quality of the output of a general-purpose, non-domain-tuned MT system.

Although the MS translations represent the most difficult set for the QE task, they are the best translations according to BLEU score (0.39 compared to 0.37 for MOSES and 0.35 for SYSTRAN). However it is not possible to conclude from this that there is a negative correlation between MT and QE performance since there are only three MT systems and the differences between them are small. Whether or not this points to a general trend requires further experimentation with other QE datasets, feature sets and MT systems.

	base.	win	full	+ full sel.	+ type sel.
MAE	0.69	0.61	0.67	0.67	0.67
RMSE	0.82	0.75	0.84	0.84	0.83

Table 6: Comparison between the baseline, the shared task winner and our approach on the WMT12 QE dataset.

To test the portability of our feature set and feature selection methods, we evaluate them on the WMT 2012 shared task dataset. The feature set contains the same features as the *Extended* set, apart from the ones which could not be extracted from the training data provided by the shared task organisers, such as forum and Symantec TM LMs and  $n$ -gram features. We report the results in Table 6. We do not outperform the baseline features in terms of RMSE but we do with MAE. Feature selection does not bring any improvement in MAE, but RMSE is slightly improved when selection is carried out at the level of individual feature type. Our system lags behind the top-ranked system (Soricut et al., 2012) – more feature selection experimentation is required in order to narrow this gap. It would also be interesting to see how the top-ranked system performs on our UGC dataset.

## 7 Conclusion

We have conducted a series of quality estimation experiments on English-French user-generated content, estimating both translation comprehensibility and fidelity, and training systems on the output of three individual MT systems. The experiments show that the information brought by a type of feature can be more or less useful depending on the MT system used. We show that the baseline suggested by the WMT12 QE shared task organisers leads to respectable results on user-generated content but we also show that there is sometimes some modest benefit to be found in extending this feature set. The features that are designed specifically to take into account that our data is user-generated content did not perform as well as other new features. However, we cannot conclude from this that modelling the forum characteristics of our data is unnecessary since the LM features trained on forum text perform well. In the future we plan to apply QE at the level of forum post rather than segment.

## Acknowledgments

The research reported in this paper has been supported by the Irish Research Council Enterprise Partnership Scheme (EPSPG/2011/102 and EPSPD/2011/135) and the computing infrastructure of the Centre for Next Generation Localisation ([www.cngl.ie](http://www.cngl.ie)) at Dublin City University.

## References

- Joshua S. Albrecht and Rebecca Hwa. 2007. Regression for Sentence-Level MT Evaluation with Pseudo References. In *ACL*, pages 296–303.
- Pratyush Banerjee, Sudip Kumar Naskar, Johann Roturier, Andy Way, and Josef van Genabith. 2012. Domain Adaptation in SMT of User-Generated Forum Content Guided by OOV Word Reduction: Normalization and/or Supplementary Data? In *EAMT*, pages 169–176.
- Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *COLING/ACL Workshop on Interactive presentation sessions*, pages 69–72.
- John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence Estimation for Machine Translation. In *JHU/CLSP Summer Workshop Final Report*.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022.
- Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *WMT*, pages 1–44.
- 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 *WMT*, pages 10–51.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIB-SVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.
- Jacques Duchateau, Kris Demuynck, and Patrick Wambacq. 2002. Confidence Scoring Based on Backward Language Models. In *ICASSP*, pages 221–224.
- Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-Level MT Evaluation Without Reference Translations: Beyond Language Modeling. In *EAMT*, pages 103–111.
- Simona Gandrabur and George Foster. 2003. Confidence Estimation for Translation Prediction. In *CoNLL*, pages 95–102.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.
- Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *ACL*, pages 423–430.
- 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 *ACL*, pages 177–180.
- Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *ACL*, pages 440–447.
- Christopher Quirk. 2004. Training a Sentence-Level Machine Translation Confidence Measure. In *LREC*, pages 825–828.
- Sylvain Raybaud, Caroline Lavecchia, David Langlois, and Kamel Smaili. 2009. Word-and Sentence-Level Confidence Measures for Machine Translation. In *EAMT*, pages 104–111.
- Sylvain Raybaud, David Langlois, and Kamel Smaïli. 2011. "This Sentence is Wrong." Detecting Errors in Machine-Translated Sentences. *Machine Translation*, pages 1–34.
- Johann Roturier and Anthony Bensadoun. 2011. Evaluation of MT Systems to Translate User Generated Content. In *MT Summit*, pages 244–251.
- Raphael Rubino, Jennifer Foster, Joachim Wagner, Johann Roturier, Rasoul Samad Zadeh Kaljahi, and Fred Hollowood. 2012. DCU-Symantec Submission for the WMT 2012 Quality Estimation Task. In *WMT*, pages 138–144.
- Raphael Rubino, José Guilherme Camargo de Souza, Jennifer Foster, and Lucia Specia. 2013. Topic Models for Translation Quality Estimation for Gisting Purposes. In *MT Summit*.
- Radu Soricut, Nguyen Bach, and Ziyuan Wang. 2012. The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task. In *WMT*, pages 145–151.
- Lucia Specia and Jesús Gimenez. 2010. Combining Confidence Estimation and Reference-Based Metrics for Segment Level MT Evaluation. In *AMTA*.
- Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *EAMT*, pages 28–35.
- Nicola Ueffing and Hermann Ney. 2004. Bayes Decision Rules and Confidence Measures for Statistical Machine Translation. *Advances in Natural Language Processing*, pages 70–81.
- Nicola Ueffing, Klaus Macherey, and Hermann Ney. 2003. Confidence Measures for Statistical Machine Translation. In *MT Summit*.

# Selective Combination of Pivot and Direct Statistical Machine Translation Models

Ahmed El Kholy, Nizar Habash

Center for Computational Learning Systems, Columbia University  
{akholy, habash}@ccls.columbia.edu

Gregor Leusch, Evgeny Matusov

Science Applications International Corporation  
{gregor.leusch, evgeny.matusov}@saic.com

Hassan Sawaf

eBay Inc.  
hsawaf@ebay.com

## Abstract

In this paper, we propose a selective combination approach of pivot and direct statistical machine translation (SMT) models to improve translation quality. We work with Persian-Arabic SMT as a case study. We show positive results (from 0.4 to 3.1 BLEU on different direct training corpus sizes) in addition to a large reduction of pivot translation model size.

## 1 Introduction

One of the main challenges in statistical machine translation (SMT) is the scarcity of parallel data for many language pairs especially when the source and target languages are morphologically rich. Morphological richness comes with many challenges and the severity of these challenges increases when the richness and morphological complexity are expressed differently in the source and target languages.

A common SMT solution to the lack of parallel data is to pivot the translation through a third language (called pivot or bridge language) for which there exist abundant parallel corpora with the source and target languages. The literature covers many pivoting techniques. One of the best performing techniques, phrase pivoting (Utiyama and Isahara, 2007), builds an induced new phrase table between the source and target. One of the problems of this technique is that the size of the newly created pivot phrase table is very large (Utiyama and Isahara, 2007).

Given a parallel corpus between the source and target language, combining a direct model based on this parallel corpus with a pivot model could lead to better coverage and overall translation quality. However, the combination approach needs

to be optimized in order to maximize the information gain.

In this paper, we propose a selective combination approach of pivot and direct SMT models. The main idea is to select the relevant portions of the pivot model that do not interfere with the more trusted direct model. We show positive results for Persian-Arabic SMT (from 0.4 to 3.1 BLEU on different direct training corpus sizes). As a positive side effect, we achieve a large reduction of pivot translation model size.

This paper is organized as follows. Section 2 briefly discusses some related work. Section 3 presents linguistic challenges and differences between Arabic and Persian. In Section 4, we discuss our pivoting strategies. Then Section 5 discusses our approach for selective combination. In Section 6, we present our experimental results.

## 2 Related Work

### 2.1 Pivoting

Many researchers have investigated the use of pivoting (or bridging) approaches to solve the data scarcity issue (Utiyama and Isahara, 2007; Wu and Wang, 2009; Khalilov et al., 2008; Bertoldi et al., 2008; Habash and Hu, 2009). The core idea is to introduce a pivot language, for which there exist large source-pivot and pivot-target bilingual corpora. Pivoting has been explored for closely related languages (Hajič et al., 2000) as well as unrelated languages (Koehn et al., 2009; Habash and Hu, 2009). Many different pivoting strategies have been presented in the literature. The following two are perhaps the most commonly used.<sup>1</sup>

<sup>1</sup>Another notable strategy is to create a synthetic source-target corpus by translating the pivot side of source-pivot corpus to the target language using an existing pivot-target model (Bertoldi et al., 2008). A new source-target model is built from the new corpus.

The first strategy is sentence pivoting in which we first translate the source sentence to the pivot language, and then translate the pivot language sentence to the target language (Khalilov et al., 2008).

The second strategy is phrase pivoting (Utiyama and Isahara, 2007; Cohn and Lapata, 2007; Wu and Wang, 2009). In phrase pivoting, a new source-target phrase table (translation model) is induced from source-pivot and pivot-target phrase tables. We compute the lexical weights and translation probabilities from the two phrase tables.

In this paper, we utilize the phrase pivoting strategy as our baseline, which is shown to be better in performance compared to sentence pivoting (El Kholy et al., 2013).

## 2.2 Domain Adaptation

We propose a selective combination approach of pivot and direct SMT models to improve the translation quality. Our approach is similar to domain adaptation techniques where training data from many diverse sources are combined to build a single translation model which is used to translate sentences in a new domain.

Domain adaptation has been explored in the field through different methods. Some methods involve information retrieval (IR) techniques to retrieve sentence pairs related to the target domain from a training corpus (Eck et al., 2004; Hildebrand et al., 2005). Other domain adaptation methods are based on distinguishing between general and domain specific examples (Daumé III and Marcu, 2006). In a similar approach, Koehn and Schroeder (2007) use multiple alternative decoding paths to combine different translation models and the weights are set with minimum error rate training (Och and Ney, 2003).

In contrast to domain adaptation, we generate a new source-target translation model by phrase pivoting technique from two models. We then use domain adaptation approach to select relevant portions of the pivot phrase table and combine them with a direct translation model to improve the overall translation quality.

## 2.3 Morphologically Rich Languages

Since both Persian and Arabic are morphologically rich, we should mention that there has been a lot of work on translation to and from morphologically rich languages (Yeniterzi and Oflazer, 2010; Elming and Habash, 2009; El Kholy and Habash, 2010a; Habash and Sadat, 2006; Kathol and Zheng, 2008; Shilon et al., 2010). Most of

these efforts are focused on syntactic and morphological processing.

There have been a growing number of publications that consider translation into Arabic. Sarikaya and Deng (2007) use joint morphological-lexical language models to re-rank the output of English-dialectal Arabic MT. Other efforts report results on the value of morphological tokenization of Arabic during training and describe different techniques for detokenizing Arabic output (Badr et al., 2008; El Kholy and Habash, 2010b).

On the other hand, work on Persian SMT is limited to few studies. For example, Kathol and Zheng (2008) use unsupervised morpheme segmentation for Persian. They show that hierarchical phrase-based models can improve Persian-English translation. There are also other attempts to improve Persian-English SMT by working on syntactic reordering (Gupta et al., 2012) and rule-based post editing (Mohaghegh et al., 2012). There is also some work done on showing the effect of different orthographic and morphological processing for Persian on Persian-English translation (Rasooli et al., 2013a).

To our knowledge, there hasn't been a lot of work on Persian and Arabic as a language pair. One example is an effort based on improving the reordering models for Persian-Arabic SMT (Matusov and Köprü, 2010). Another recent effort improved the quality of Persian-Arabic by pivoting through English and adding additional features to reflect the quality of projected alignments between the source and target phrases in the pivot phrase table (El Kholy et al., 2013).

## 3 Arabic and Persian Linguistic Issues

In this section we present our motivation and choice for preprocessing Arabic, Persian and English data. Both Arabic and Persian are morphologically complex languages but they belong to two different language families. They both express richness and linguistic complexities in different ways (El Kholy et al., 2013).

One aspect of Arabic's complexity is its various attachable clitics and numerous morphological features (Habash, 2010) which include conjunction proclitics, e.g.,  $+و$  *w+* 'and', particle proclitics, e.g.,  $+ل$  *l+* 'to/for', the definite article  $+ال$  *Al+* 'the', and the class of pronominal enclitics, e.g.,  $+هم$  *+hm* 'their/them'. Beyond these clitics, Arabic words inflect for person, gender, number,

aspect, mood, voice, state and case.<sup>2</sup> This morphological richness leads to thousands of inflected forms per lemma and a high degree of ambiguity: about 12 analyses per word, typically corresponding to two lemmas on average (Habash, 2010). We follow El Kholy and Habash (2010a) and use the PATB tokenization scheme (Maamouri et al., 2004) in our experiments which separates all clitics except for the determiner clitic *Al+*. We use MADA v3.1 (Habash and Rambow, 2005; Habash et al., 2009) to tokenize the Arabic text. We only evaluate on detokenized and orthographically correct (enriched) output following the work of El Kholy and Habash (2010b).

Persian on the other hand has a relatively simple nominal system. There is no case system and words do not inflect with gender except for a few animate Arabic loanwords. Unlike Arabic, Persian shows only two values for number, just singular and plural (no dual), which are usually marked by either the suffix *ها+ +hA* and sometimes *ان+ +An*, or one of the Arabic plural markers. Persian also possess a closed set of few broken plurals loaned from Arabic. Moreover, unlike Arabic which expresses definiteness, Persian expresses indefiniteness with an enclitic article *ی+ +y* ‘*a/an*’ which doesn’t have separate forms for singular and plural. When a noun is modified by one or more adjective, the indefinite article is attached to the last adjective. Persian adjectives are similar to English in expressing comparative and superlative constructions just by adding suffixes *تر+ +tar* ‘+er’ and *ترین+ +taryn* ‘+est’ respectively. Verbal morphology is very complex in Persian. Each verb has a past and present root and many verbs have attached prefix that is regarded part of the root. A verb in Persian inflects for 14 different tense, mood, aspect, person, number and voice combination values (Rasooli et al., 2013b).

We follow El Kholy et al. (2013) and tokenize Persian text using Perstem (Jadidinejad et al., 2010) which is a deterministic rule based approach for segmentation of Persian.

English, our pivot language, is quite different from both Arabic and Persian. English is poor in morphology and barely inflects for number and tense, and for person in a limited context. English preprocessing simply includes down-casing, separating punctuation and splitting off “’s”.

<sup>2</sup>We use the Habash-Soudi-Buckwalter Arabic transliteration (Habash et al., 2007) with extensions for Persian as suggested by Habash (2010).

## 4 Pivoting Strategies

In this section, we review the two pivoting strategies that are our baselines.

### 4.1 Sentence Pivoting

In sentence pivoting, English is used as an interface between two separate phrase-based MT systems; Persian-English direct system and English-Arabic direct system. Given a Persian sentence, we first translate the Persian sentence from Persian to English, and then from English to Arabic.

### 4.2 Phrase Pivoting

In phrase pivoting (sometimes called triangulation or phrase table multiplication), we train a Persian-to-Arabic and an English-Arabic translation models, such as those used in the sentence pivoting technique. Based on these two models, we induce a new Persian-Arabic translation model.

Since we build our models are based Moses phrase-based SMT (Koehn et al., 2007), we provide the basic set of phrase translation probability distributions.<sup>3</sup> We follow Utiyama and Isahara (2007) in computing the probability distributions. The following are the set of equations used to compute the lexical probabilities ( $\phi$ ) and the phrase translation probabilities ( $p_w$ )

$$\begin{aligned}\phi(f|a) &= \sum_e \phi(f|e)\phi(e|a) \\ \phi(a|f) &= \sum_e \phi(a|e)\phi(e|f) \\ p_w(f|a) &= \sum_e p_w(f|e)p_w(e|a) \\ p_w(a|f) &= \sum_e p_w(a|e)p_w(e|f)\end{aligned}$$

where  $f$  is the Persian source phrase.  $e$  is the English pivot phrase that is common in both Persian-English translation model and English-Arabic translation model.  $a$  is the Arabic target phrase.

We also build a Persian-Arabic reordering table using the same technique but we compute the reordering probabilities in a similar manner to Henriquez et al. (2010).

As discussed earlier, the induced Persian-Arabic phrase and reordering tables are very large. Table 1 shows the amount of parallel corpora used to train the Persian-English and the English-Arabic and the equivalent phrase table sizes compared to the induced Persian-Arabic phrase table.<sup>4</sup>

<sup>3</sup>Four different phrase translation scores are computed in Moses’ phrase tables: two lexical weighting scores and two phrase translation probabilities.

<sup>4</sup>The size of the induced phrase table size is computed but not created.

Translation Model	Training Corpora Size	Phrase Table	
		# Phrase Pairs	Size
Persian-English	≈4M words	96,04,103	1.1GB
English-Arabic	≈60M words	111,702,225	14GB
Pivot_Persian-Arabic	N/A	39,199,269,195	≈2.5TB

Table 1: Translation Models Phrase Table comparison in terms of number of line and sizes.

We follow the work of El Kholy et al. (2013) and filter the phrase pairs used in pivoting based on log-linear scores. We present some baseline results to show the effect of filtering on the translation quality in Section 6.2.

## 5 Approach

In this section, we discuss our selective combination approach. We explore how to effectively combine both a pivot and a direct model built from a given parallel corpora to achieve better coverage and overall translation quality. We maximize the information gain by selecting the relevant portions of the pivot model that do not interfere with the more trusted direct model.

To achieve this goal, we investigate the idea of classifying the pivot phrase pairs into five different classes based on the existence of source and/or target phrases in the direct model. The first class contains the phrase pairs where the source and target phrases are in the direct system together. The second class is the same as the first class except that the source and target phrases exist but not together as a phrase pair in the direct system. The third, fourth and fifth classes are for the existence of source phrase only, target phrase only and neither in the direct system. Table 2 shows the different classifications of the portions extracted from the pivot phrase table with their labels which are used later in our results tables.

We use one of Moses phrase table combination techniques (Koehn and Schroeder, 2007) to combine the direct model with the different pivot portions (explained in more details in section 6.1).

## 6 Experiments

In this section, we present our results for the selective combination approach between direct and pivoting models.

### 6.1 Experimental Setup

For the direct Persian-Arabic SMT model, we use an inhouse parallel corpus of about 165k sentences and 4 million words.

Pivot phrase-pairs classification	Src exists in direct	Tgt exists in direct	Src & Tgt exist in direct
SRC : TGT	✓	✓	✓
SRC , TGT	✓	✓	×
SRC ONLY	✓	×	×
TGT ONLY	×	✓	×
NEITHER	×	×	×

Table 2: Phrase pairs classification of the portions extracted from the pivot phrase table.

In our pivoting experiments, we build two SMT models. One model to translate from Persian to English and another model to translate from English to Arabic. The English-Arabic parallel corpus is about 2.8M sentences (≈60M words) available from LDC<sup>5</sup> and GALE<sup>6</sup> constrained data. We use an in-house Persian-English parallel corpus of about 170K sentences and 4M words.

Word alignment is done using GIZA++ (Och and Ney, 2003). For Arabic language modeling, we use 200M words from the Arabic Gigaword Corpus (Graff, 2007) together with the Arabic side of our training data. We use 5-grams for all language models (LMs) implemented using the SRILM toolkit (Stolcke, 2002). For English language modeling, we use the English Gigaword Corpus with 5-gram LM using the KenLM toolkit (Heafield, 2011).

All experiments are conducted using Moses phrase-based SMT system (Koehn et al., 2007). We use MERT (Och, 2003) for decoding weights optimization. For Persian-English translation model, weights are optimized using a set 1000 sentences randomly sampled from the parallel corpus while the English-Arabic translation model weights are optimized using a set of 500 sentences from the 2004 NIST MT evaluation test set

<sup>5</sup>LDC Catalog IDs: LDC2005E83, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006G05, LDC2007E06, LDC2007E101, LDC2007E103, LDC2007E46, LDC2007E86, LDC2008E40, LDC2008E56, LDC2008G05, LDC2009E16, LDC2009G01.

<sup>6</sup>Global Autonomous Language Exploitation, or GALE, is a DARPA-funded research project.

(MT04).

We use a maximum phrase length of size 8 across all models. We report results on an in-house Persian-Arabic evaluation set of 536 sentences with three references. We evaluate using BLEU-4 (Papineni et al., 2002).

For the combination experiments, Moses allows the use of multiple translation tables (Koehn and Schroeder, 2007). Different combination techniques are available. We use the ‘‘Either’’ combination technique where the translation options are collected from one table, and additional options are collected from the other tables. If the same translation option (identical source and target phrases) is found in multiple tables, separate translation options are created for each occurrence, but with different scores.

## 6.2 Baseline Evaluation

We compare the performance of sentence pivoting against phrase pivoting with different filtering thresholds. The results are presented in Table 3. In general, the phrase pivoting outperforms the sentence pivoting even when we use a small filtering threshold of size 100. Moreover, the higher the threshold the better the performance but with a diminishing gain.

Pivot Scheme	BLEU
Sentence Pivoting	19.2
Phrase_Pivot_F100	19.4
Phrase_Pivot_F500	20.1
Phrase_Pivot_F1K	<b>20.5</b>

Table 3: Sentence pivoting versus phrase pivoting with different filtering thresholds (100/500/1000).

We use the best performing setup across the rest of the experiments which is filtering with a threshold of 1K.

## 6.3 System Combinations

In this section, we investigate the selective combination approach. We start by the basic combination approach and then explore the gain/loss achieved from dividing the pivot phrase table to five different classes as discussed in Section 5.

### 6.3.1 Baseline Combination

Table 4 shows the results of the basic combination in comparison to the best pivot translation model and the best direct model. The results shows that combining both models leads to a gain in performance. The question is how to improve the quality

by doing a smart selection of only relevant portion of the pivot phrase table which is discussed next.

Model	BLEU%
Phrase_Pivot_F1K	20.5
Direct	23.4
Direct+Phrase_Pivot_F1K	<b>23.7</b>

Table 4: Baseline combination experiments between best pivot baseline and best direct model.

### 6.3.2 Selective Combination

In this section, we explore the idea of dividing the pivot phrase pairs into five different classes based on the existence of source and/or target phrases in the direct system as discussed in Section 5. We discuss our results and show the trade off between the quality of translation and the size of the different classes extracted from the pivot phrase table.

Table 5 shows the results of the selective combination experiments on a learning curve of 100% (4M words), 25% (1M words) and 6.25% (250K words) of the parallel Persian-Arabic corpus.

Model	Parallel data set size		
	4M	1M	250K
Direct	23.4	21.0	16.8
Phrase_Pivot_F1K	20.5		
Base Combination	23.7 *	<b>22.1 *</b>	<b>21.7 *</b>
SRC : TGT	22.9	21.2	17.3 *
SRC , TGT	23.0	21.3	18.5 *
SRC ONLY	23.5	20.1	17.5 *
TGT ONLY	<b>23.8*</b>	21.4 *	18.3 *
NEITHER	23.4	21.6 *	19.9 *

Table 5: Selective Combination experiments results on a learning curve. The first row shows the results of the direct system. The second row shows the result of the best pivot system. The third row shows the results of the baseline combination experiments with the whole pivot phrase table. Then the next set of rows show the results of the selective combination experiments based on the different classifications. All scores are in BLEU. (\*) marks a statistically significant result against the direct baseline.

The results show that pivoting is a robust technique when there is no or small amount of parallel corpora. In our case study on Persian-Arabic SMT, the direct translation systems built from parallel corpora starts to be better than the pivot translation system when trained on 1M words or more.

The base combination between the direct translation models and the pivot translation model leads to a boost in the translation quality across the learning curve. As expected, the smaller the parallel corpus used in training the more gain we get from the combination.

The results also show that some of pivot the classes provides more information gain than others. In fact some of the classes hurt the overall quality; for example, (SRC : TGT) and (SRC , TGT) both hurt the quality of translation when combined with direct model trained on 100% of the parallel data (4M words).

An interesting observation from the results is that by building a translation system with only 6.25% of the parallel data ( $\approx$  250K words) combined with the pivot translation model, we can achieve a better performance (21.7 BLEU) than a model trained on four times the amount of data (Size: 1M words; Score: 21.0 BLEU).

It is also shown across the learning curve that the best gains are achieved when the source phrase in the pivot phrase table doesn't exist in the direct model. This is expected due to the fact that by adding unknown source phrases, we decrease the overall OOVs.

Model	Parallel data set size		
	4M	1M	250K
SRC : TGT	0.2%	0.1%	0.1%
SRC , TGT	35.2%	29.0%	16.0%
SRC ONLY	59.9%	63.3%	64.1%
TGT ONLY	2.3%	3.4%	6.1%
NEITHER	2.3%	4.3%	13.7%

Table 6: Percentage of phrase pairs extracted from the original pivot phrase table for each pivot class across the learning curve.

Pruning the pivot phrase table is an additional benefit from the selective combination approach. Table 6 shows that percentage of phrase pairs extracted from of the original pivot phrase table for each pivot class across the learning curve. The bulk of the phrase pairs are extracted in the classes where the source phrases exist in the direct model which add the least and sometimes hurt the overall combination performance.

For the large parallel data (4M words), selective combination with (TGT ONLY) class gives a slightly better result in BLEU while hugely reducing the size of the pivot phrase table used (2.3% of the original pivot phrase table). For smaller parallel data, the advantage is reduced but here comes

the trade off between the quality of the translation and the size of the model.

## 7 Conclusion and Future Work

We propose a selective combination approach between pivot and direct models to improve the translation quality. We showed that the selective combination can lead to a large reduction of the pivot model without affecting the performance if not improving it. In the future, we plan to investigate classifying the pivot model based on morphological patterns extracted from the direct model instead of just the exact surface form.

## Acknowledgments

The work presented in this paper was possible thanks to a generous research grant from Science Applications International Corporation (SAIC). The last author (Sawaf) contributed to the effort while he was at SAIC. We would like to thank M. Sadeqh Rasooli, Jon Dehdari and Nadi Tomeh for helpful discussions and insights into Persian.

## References

- Ibrahim Badr, Rabih Zbib, and James Glass. 2008. Segmentation for English-to-Arabic Statistical Machine Translation. In *Proc. of ACL'08*.
- Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. 2008. Phrase-based statistical machine translation with pivot languages. *Proc. of IWSLT'08*.
- Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proc. of ACL'07*.
- Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.(JAIR)*.
- Matthias Eck, Stephan Vogel, and Alex Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In *Proc. LREC'04*.
- Ahmed El Kholy and Nizar Habash. 2010a. Orthographic and Morphological Processing for English-Arabic Statistical Machine Translation. In *Proc. of TALN'10*.
- Ahmed El Kholy and Nizar Habash. 2010c. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proc. of LREC'10*.
- 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 *Proc. of ACL'13*.



- Jakob Elming and Nizar Habash. 2009. Syntactic Reordering for English-Arabic Phrase-Based Machine Translation. In *Proc. of EACL'09*.
- David Graff. 2007. Arabic Gigaword 3, LDC Catalog No.: LDC2003T40. Linguistic Data Consortium, University of Pennsylvania.
- Rohit Gupta, Raj Nath Patel, and Ritnesh Shah. 2012. Learning improved reordering models for Urdu, Farsi and Italian using SMT. In *Proc. of WMT'12*.
- Nizar Habash and Jun Hu. 2009. Improving Arabic-Chinese Statistical Machine Translation using English as Pivot Language. In *Proc. of EACL'09*.
- Nizar Habash and Owen Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proc. of ACL'05*.
- Nizar Habash and Fatiha Sadat. 2006. Arabic Pre-processing Schemes for Statistical Machine Translation. In *Proc. of NAACL'06*.
- Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.
- Nizar Habash, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In Khalid Choukri and Bente Maegaard, editors, *Proc. of the Second International Conference on Arabic Language Resources and Tools*.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool.
- Jan Hajič, Jan Hric, and Vladislav Kubon. 2000. Machine Translation of Very Close Languages. In *Proc. of ANLP'00*.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proc. of WMT'11*.
- Carlos Henriquez, Rafael E. Banchs, and José B. Mariño. 2010. Learning reordering models for statistical machine translation with a pivot language.
- Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proc. of EAMT'05*.
- Amir Hossein Jadidinejad, Fariborz Mahmoudi, and Jon Dehdari. 2010. Evaluation of PerStem: a simple and efficient stemming algorithm for Persian. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*.
- Andreas Kathol and Jing Zheng. 2008. Strategies for building a Farsi-English smt system from limited resources. In *Proc. of INTERSPEECH'08*.
- M. Khalilov, Marta R. Costa-juss, Jos A. R. Fonollosa, Rafael E. Banchs, B. Chen, M. Zhang, A. Aw, H. Li, Jos B. Mario, Adolfo Hernandez, and Carlos A. Henriquez Q. 2008. The talp & i2r smt systems for iwslt 2008. In *Proc. of IWSLT'08*.
- Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proc. of WMT'07*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proc. of ACL'07*.
- Philipp Koehn, Alexandra Birch, and Ralf Steinberger. 2009. 462 machine translation systems for europe. *Proc. of MT Summit XII*.
- Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR'04*.
- Evgeny Matusov and Selçuk Köprü. 2010. Improving reordering in statistical machine translation from farsi. In *Proc. of AMTA'10*.
- Mahsa Mohaghegh, Abdolhossein Sarrafzadeh, and Mehdi Mohammadi. 2012. GRAFIX: Automated rule-based post editing system to improve English-Persian SMT output. In *Proc. of COLING'12*.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL'03*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL'02*.
- Mohammad Sadegh Rasooli, Ahmed El Kholy, and Nizar Habash. 2013a. Orthographical and morphological processing for persian to english statistical machine translation,. In *Proc. of IJCNLP'13*.
- Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. 2013b. Development of a Persian syntactic dependency treebank. In *Proc. of NAACL'13*.
- Ruhi Sarikaya and Yonggang Deng. 2007. Joint morphological-lexical language modeling for machine translation. In *Proc. of NAACL'07*.
- Reshef Shilon, Nizar Habash, Alon Lavie, and Shuly Wintner. 2010. Machine Translation between Hebrew and Arabic: Needs, Challenges and Preliminary Solutions. In *Proc. of AMTA'10*.
- Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proc. of ICSLP'02*.
- Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Proc. of NAACL'07*.
- Hua Wu and Haifeng Wang. 2009. Revisiting pivot language approach for machine translation. In *Proc. of ACL'09*.
- Reyyan Yeniterzi and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from english to turkish. In *Proc. of ACL'10*.

# Multiword Expressions in the Context of Statistical Machine Translation

**Mahmoud Ghoneim**

Center for Computational Learning Systems  
Columbia University  
475 Riverside Drive MC 7717  
New York, NY 10115  
mah.ghoneim@gmail.com

**Mona Diab**

Department of Computer Science  
George Washington University  
801 22nd Street NW  
Washington DC 20052  
mtdiab@email.gwu.edu

## Abstract

Incorporating semantic information in the Statistical Machine Translation (SMT) framework is starting to gain some popularity in both the semantics and translation communities. In this paper, we present encouraging results obtained from experiments conducted on English to Arabic SMT system using static, dynamic, and hybrid integration of fine-grained Multiword Expression (MWE). We achieve an improvement up to 0.82 absolute BLEU score by integrating MWEs over a vanilla SMT system. We empirically show that different MWE types require different integration methods in the SMT framework.

## 1 Introduction

Multiword expressions (MWEs) are roughly defined by (Sag et al., 2002) as “idiosyncratic concepts that cross word boundaries (spaces).” MWEs are widely used, 41% of the entries in WordNet 1.7 (Fellbaum, 1998) are MWEs, but unfortunately they have proved to be hard to model in natural language processing applications. Typical statistical machine translation (SMT) systems, in particular, do not explicitly model MWEs. This might indicate that state of the art SMT systems are doing well without having any knowledge of whether a given phrase is a multiword expression or not. However, recent research (Carpuat and Diab 2010, Bouamor et al., 2012) show that explicitly modeling MWEs in the SMT framework yields non-negligible gains depending on the integration method.

In this paper we study explicit modeling of the diverse kinds of MWEs in a phrase-based SMT framework for the English-Arabic language pair. This paper is organized as follows: section 2 overviews the different types of MWEs, section

3 reviews the previous work related to MWEs and SMT. Section 4 details our approach followed by the results in section 5. Our discussion of the results is presented in section 6 and finally the conclusions are in section 7.

## 2 Multiword Expressions Classification

According to (Sag et al., 2002), MWEs are broadly classified into institutionalized phrases and lexicalized phrases based on the varying degree of lexical rigidity and semantic compositionality.

**Institutionalized phrases** are conventionalized phrases that are syntactically and semantically compositional, but statistically idiosyncratic (e.g. “traffic light”, “to kindle excitement”).

**Lexicalized phrases** have at least in part idiosyncratic syntax or semantics. They can be further broken down into:

(a) **Fixed expressions** which undergo neither morphosyntactic variation, nor internal modification (e.g. “by and large”, “every which way”) [AV, AJ],

(b) **Semi-fixed expressions** such as (1) non-decomposable idioms (e.g. “kick the bucket”) [VNC], (2) compound nominal (e.g. “car park”, “part of speech”) [NNC], and (3) proper names and named entities (e.g. “New York”) [NE].

(c) **Syntactically-flexible expressions** such as (1) verb particle construction (e.g. “write up”, “look up”) [VPC], (2) light verb constructions (e.g. “make a decision”) [LVC], and (3) decomposable idioms (e.g. “sweep under the rug”) [VNC].

## 3 Related Work

Previous work has focused on automatically learning and integrating translations of very specific MWE categories, such as, for instance, idiomatic Chinese four character expressions (Bai

et al., 2009.) MWEs have also been defined not from a lexical semantics perspective but from a SMT error reduction perspective, as phrases that are hard to align during SMT training (Lambert and Banchs, 2005). For each of these particular cases, translation quality improved by augmenting the SMT translation lexicon with the learned bilingual MWEs either directly or through improved word alignments.

Ren et al. (2009) described a method integrating an in-domain bilingual MWE to Moses by introducing an additional feature that identifies whether or not a bilingual phrase contains bilingual MWEs. This approach was generalized in Carpuat and Diab (2010) who replaced the binary feature by a count feature representing the number of MWEs in the source language phrase. They present results on a large data set of English to Arabic SMT. They introduce two ways of integrating MWE knowledge in the SMT framework: Static and Dynamic integration. For Static integration, MWE tokens in the source data are grouped together with an underscore. While in Dynamic integration, the MWEs are identified in the phrase table and an additional weighted feature, as a soft constraint, is added to the phrase translation table. Carpuat and Diab (2010) focus only on MWEs as identified in WordNet (Fellbaum, 1998) with no explicit distinction between the different types of MWEs. Accordingly, the MWEs are considered a single type with no attention to various POS information. Our work here is taking a much fine grained approach and deeper study and analysis.

## 4 Approach

We adopt a Phrase-based SMT framework, Moses (Koehn et al., 2007). In the following subsections, we address the issue of representation of MWE in our SMT pipeline and then we investigate the manner in which the MWE information is integrated in the SMT framework.

### 4.1 Data Sets

For training the translation models, we use LDC GALE newswire parallel Arabic-English corpus (LDC2007E103) (a total of 474299 sentence pairs / about 10M un-tokenized words / 12M tokenized words). The Log-Linear model features weights are tuned using the newswire part of NIST MT06 (765 sentence pairs) as the tuning dataset and BLEU (Papineni et al., 2002) as the objective function. For training the language model (LM), we use the LDC Arabic

GIGAWORD 4th edition (LDC2009T30) (about 850M un-tokenized words).

We use the newswire part of NIST-MT04 (707 sentences) as our development test-set to compare performance and select combinations of different conditions. We report results using two blind test-sets; NIST-MT05 (1056 sentences) and the newswire part of NIST-MT08 (813 sentences). These standard test sets are originally designed to test Arabic to English translation systems thus it consists of one Arabic source set and four English human reference translation sets. To use these test sets for testing English to Arabic translation systems, we created new test sets where the source set is constructed by concatenating the four English human translations of the original standard test set, and the reference set is constructed by duplicating the original standard test set Arabic source four times. This means that the new test sets have four times the number of sentences of the original standard test sets. Increasing the test set size enhances the reliability of the evaluation scores as reported by (Zhang and Vogel 2010).

### 4.2 MWEs lists

We need a mechanism by which to identify MWE in the source English text. We rely on two identification sources depending on the type of MWE: an MWE list extracted from a wide coverage lexical database and a named entity recognition tool. As mentioned earlier in section 2, we consider several types of MWEs for this study: Verb-based MWEs (VNC, VPC, and LVC), Noun-based MWEs (NNC, and NE), Adjective (AJ) and Adverb (AV) based MWE.

#### WordNet Extracted MWEs Lists:

For the VPC, VNC, LVC, NNC and AJ and AV categories of MWE, we extract an extensive list from the wide coverage English WordNet database 3.0. (Fellbaum,1998). Table 1 shows the number of MWEs extracted from WordNet 3.0 dictionaries. It is worth noting that the MWE.V list comprises all three types of verbal MWEs (VNC, VPC, LVC), moreover the MWE.N includes NNC and some NEs as listed in WordNet.

MWE list	# MWE types
MWE.V	3,089
MWE.N	62,244
MWE.AJ	3,358
MWE.AV	826

Table 1: WordNet 3.0 based MWE statistics

## Named Entities Tagging:

We consider Named Entities (NEs) as another type of MWE. To construct our NEs list, we exploit a named entity tagger, the Stanford NER [SNER] (Finkel et al., 2005). SNER tags named entities in a given English text into three categories: 1) Person 2) Organization and 3) Location. We are interested in Multiword NEs only and pay no attention to the different NE categories. The extracted NEs list consists of the 65616 Multiword NEs tagged by SNER in our training corpus.

There are some overlaps between the NEs list and the MWE lists extracted from WordNet as shown in table 2. The large overlap is between the NEs list and the MWE.N, which contains NEs as listed in WordNet 3.0.

	MWE.N	MWE.AJ	MWE.AV
# types	1216	24	5
Examples	abraham lincoln abu dhabi abu sayyaf adam smith addis ababa adriatic sea	african american anti american central american costa rican east african eastern orthodox	north east north northeast north west south east south west

Table 2: Overlaps between the WordNet MWEs lists and the NEs list

## Matching Algorithm:

In order to identify the MWE in the source English side of the parallel data, we use a Maximum Forward Matching algorithm that finds the longest matching MWE in the text. The algorithm matches over the tokenized version of the data and if no match, it backs-off to the lemmatized version to account for the different inflectional forms of the MWE (e.g. “take place” and “took place”). Our current matching algorithm doesn’t handle gap flexibility like in the phrasal verbs MWEs (i.e. “break up” is handled while “break it up” is not.)

## 4.3 SMT System

### Data preprocessing and models generation:

The Arabic side of the train, tune, development and test data sets and the language model training data sets are tokenized using AMIRA 2.1 toolkit (Diab 2009, Diab et al., 2007) into the Arabic TreeBank tokenization scheme. The Arabic side of the training data is further processed to generate a lemmatized version used in the alignment stage of the SMT pipeline. We use the undiacritized version (both tokenized and lemmatized) in all our experiments.

The English side is tokenized using Tree Tagger (Schmid, 1994). It is then tagged using the selected MWE list according to the condition under investigation. The English lemmatized version of the training data is also generated for use in alignment.

We used SRILM toolkit (Stolcke, 2002) to create a 5-gram Arabic LM modified using Kneser-Ney smoothing.

In all our experimental conditions, the parallel corpus is word-aligned using GIZA++ in both translation directions using the lemmatized version of both sides to decrease data sparseness, and phrase translations of up to 10 words are extracted from the tokenized version of both sides using the grow-diag-final-and heuristic (Koehn et al., 2007).

We optimized log-linear model feature weights using Minimum Error Rate Training (MERT) (Och, 2003). To account for the instability of MERT, we run the tuning step three times per condition with different random seeds and use the optimized weights that give the median score.

## Integration Methods:

### (a) Static Integration (S)

In Static integration of MWEs in SMT, MWEs in English training, tuning and testing data are underscored as a preprocessing step based on a pattern match to the WN list entries and NER results. Hence static integration is a manipulation on the data representation, the SMT system is kept intact.

### (b) Dynamic Integration (D)

Dynamic integration is a soft constraint strategy that adds a new feature into the log linear model of phrase-based SMT. It is a count feature indicating the number of MWEs in the English phrase in the phrase table, thereby biasing the system, at decoding time, towards using phrases that do not break MWEs. The training, tuning, development and test data do not undergo any MWEs annotation (no underscoring).

### (c) Zone Integration (Z)

We define constrained reordering zones for all MWEs found in the test data and the decoder is forced to respect these boundaries while constructing the translation hypothesis. This is easily represented using XML tags in the system input to Moses decoder (Koehn and Haddow, 2009). It is worth noting that words within a zone are not necessarily translated as a single phrase and can be reordered; input phrases that cross zone

boundaries can be used in translation hypotheses without breaking the reordering constraint.

#### (d) *Hybrid Integration*

Motivated by the development-set results of the previous integration methods and MWEs schemes, we carried out a set of experiments investigating combining the best performing conditions.

#### **MWEs Schemes:**

We created 7 MWEs schemes combining the various types of WordNet-based MWE lists and NEs list. They are listed in Table 3, along with the number of types and tokens of MWEs found in the training data according to each of the MWE Schemes.

We combine MWEs schemes and integration methods to get the different experimental conditions listed in Table 4. Here is some example input preprocessing for the same sentence according to different conditions:

-Baseline (and all dynamic integration):

invading iraqis kurdistan is no longer an easy task .

-S\_VAA<sup>1</sup>:

invading iraqis kurdistan is no\_longer an easy task .

-S\_NN:

invading iraqis\_kurdistan is no longer an easy task .

-Z\_VAA+NN:

invading <zone> iraqis kurdistan </zone> is <zone>  
no longer </zone> an easy task .

## **5 Evaluation Results**

We used four standard MT metrics<sup>2</sup>; BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR<sup>3</sup> (Banerjee and Lavie, 2005), and TER (Snover et al., 2006), to report and compare performance of different experimental conditions. Table 4, summarizes the results.

The results show that, for the three integration methods (S, D and Z), the only conditions that help across all test-sets are S\_VAA and D\_VAA. S\_VAA gives the best results except for METEOR where D\_NN and D\_NE are outperforming S\_VAA.

<sup>1</sup> We use the convention: IntegrationMethod\_MWEScheme [-IntegrationMethod\_MWEScheme]\* to label different conditions: e.g “S\_VAA-D\_NE+NN” refers to a hybrid integration where the “VAA” MWEs are statically integrated and the “NE+NN” MWEs are dynamically integrated.

<sup>2</sup> We report case-sensitive scores as our system output is in Buckwalter transliteration.

<sup>3</sup> For METEOR scores, we used “exact” module only.

We want to investigate which part of the SMT pipeline does S\_VAA condition help, so we carried another experiment (A\_VAA) where the VAA is used in the alignment stage of the pipeline only. We simply removed underscores from the input phrases in the phrase table and the lexical reordering table and used the new tables as A\_VAA tables. The tune and test data-sets are the same as the normal baseline (no underscoring). The results show that the major part of the S\_VAA configuration enhancement is actually coming from the alignment stage.

Motivated by the development set results of S\_VAA, A\_VAA and the enhancement of METEOR scores by D\_NN and D\_NE, we carried out a couple of experiments investigating hybrids of the integration methods.

**S\_VAA-\***: In these configurations we use static integration for VAA and dynamic integration for NE and/or NN. For example, for S\_VAA-D\_NE the input phrases in the phrase table have VAA MWEs underscored and the probabilities have the added extra feature counting NEs in the input phrase. The train, tune and test data for this configuration has VAA MWEs underscored.

**A\_VAA-\***: In these configurations we use the phrase tables of the S\_VAA and remove underscores from the input phrases. We then add the extra feature indicating the counts of the NE and/or NN MWEs found in the input phrase.

Table 4 shows that A\_VAA-D\_NE+NN gives the best overall consistent performance with absolute BLEU score improvement of 0.63 for MT04-NW, 0.82 for MT05 and 0.45 for MT08-NW.

## **6 Discussion**

Static integration mainly helps when the MWE is a fixed expression (AV, AJ) that needs to be translated as a whole non-compositionally. That’s why we see the VAA condition (more than half of its list is fixed MWEs) giving the best results. Static integration also helps for semi-fixed expressions (VNC, NNC and NEs) conditioned by having enough training samples otherwise we increase OOV. If we look at table 3, we can see that the average number of tokens per type for all NE conditions is very low. This is mainly due to the huge number of NE types. That’s why NEs schemes do not show any improvement using static integration. On the other hand, S\_NN shows some inconsistent improvements depending on the data sparsity. For example, in our sample test sentence, S\_NN condition

created the new token “iraqis\_kurdistan” which is not in the training data.

Dynamic integration helps solving this data sparsity issue by introducing a new feature that is weighted globally using all evidences belonging

to the same category to favor phrase pairs with unbroken MWE of that category. That’s why we see some improvements for NEs and NNs in addition to VAA.

MWE Scheme	MWE List	#Lemma Types	# Token Types	# Tokens	(Tokens/Types)
NN	MWE.N	8,075	10,503	329,116	31.33
VAA	MWE.V, MWE.AJ , MWE.AV	3,003	5,733	184,899	32.25
VAA+NN	VAA, MWE.N	10,698	15,571	494,528	31.76
NE	NE	65,634	65,616	290,564	4.43
NE +NN	MWE.N, NE	72,308	74,674	502,782	6.73
NE+VAA	VAA, NE	68,600	71,308	472,718	6.63
NE+VAA+NN	VAA+NN, NE	74,915	79,728	667,686	8.37

Table 3. MWEs Schemes Statistics

Experiments	Development Set MT04-NW				Blind Test Set MT05				Blind Test Set MT08-NW			
	BLEU	NIST	MET	TER	BLEU	NIST	MET	TER	BLEU	NIST	MET	TER
Baseline	41.28	8.24	59.58	44.43	38.65	8.17	56.60	47.49	33.82	7.45	53.51	53.84
S_NE	37.54	7.57	56.59	46.99	35.86	7.56	54.08	49.67	31.57	7.00	51.64	55.49
S_NE+NN	36.67	7.39	56.82	48.23	35.05	7.39	54.49	50.78	30.37	6.79	51.66	57.23
S_NE+VAA	37.90	7.62	56.48	46.41	36.31	7.61	54.05	49.02	32.10	7.07	51.69	54.51
S_NE+VAA+NN	37.85	7.57	56.49	46.81	35.80	7.55	53.89	49.59	31.28	6.96	51.32	55.62
S_NN	40.87	8.12	59.74	45.13	38.90	8.11	57.07	47.82	33.26	7.33	53.59	54.71
S_VAA	41.82	8.30	59.77	44.01	39.47	8.27	57.14	46.71	33.99	7.51	53.76	53.28
S_VAA+NN	41.16	8.17	59.69	44.56	38.94	8.12	56.98	47.42	33.12	7.33	53.45	54.44
D_NE	41.07	8.16	<b>60.05</b>	44.74	38.89	8.12	57.18	47.57	33.33	7.36	53.85	54.34
D_NE+NN	40.86	8.16	59.69	44.78	38.74	8.11	56.94	47.63	33.56	7.38	53.72	54.18
D_NE+VAA	40.80	8.15	59.56	44.91	38.83	8.11	56.96	47.70	33.37	7.36	53.62	54.35
D_NE+VAA+NN	41.00	8.16	59.50	44.59	39.04	8.14	56.88	47.30	33.72	7.40	53.66	53.81
D_NN	41.33	8.20	<b>60.05</b>	44.45	39.20	8.15	57.27	47.29	33.66	7.39	53.73	54.10
D_VAA	41.36	8.24	59.64	44.33	38.83	8.18	56.72	47.33	33.94	7.46	53.55	53.76
D_VAA+NN	41.12	8.20	59.70	44.58	39.06	8.17	57.02	47.29	33.66	7.41	53.69	53.99
Z_NE	41.15	8.23	59.48	44.53	38.61	8.16	56.57	47.53	33.83	7.45	53.52	53.81
Z_NE+NN	41.12	8.23	59.49	44.54	38.59	8.16	56.56	47.53	33.82	7.45	53.52	53.80
Z_NE+VAA	41.13	8.23	59.45	44.53	38.60	8.16	56.57	47.52	33.78	7.45	53.51	53.83
Z_NE+VAA+NN	41.11	8.23	59.46	44.53	38.60	8.16	56.56	47.53	33.78	7.45	53.50	53.82
Z_NN	41.25	8.24	59.59	44.43	38.61	8.16	56.58	47.50	33.80	7.44	53.49	53.85
Z_VAA	41.24	8.24	59.53	44.43	38.64	8.17	56.59	47.48	33.78	7.45	53.51	53.85
Z_VAA+NN	41.22	8.24	59.54	44.43	38.62	8.16	56.58	47.49	33.76	7.44	53.49	53.86
A_VAA	41.43	8.22	59.96	44.29	39.66	8.21	<b>57.42</b>	46.85	33.96	7.45	54.03	53.54
A_VAA-D_NE	41.85	8.29	59.95	43.80	<b>39.73</b>	8.28	57.34	46.50	34.15	7.50	53.86	53.09
A_VAA-D_NE+NN	<b>41.91</b>	<b>8.37</b>	59.63	<b>43.38</b>	39.47	<b>8.35</b>	57.08	<b>46.03</b>	<b>34.27</b>	<b>7.61</b>	53.78	<b>52.43</b>
A_VAA-D_NN	41.63	8.25	59.79	44.16	39.64	8.25	57.29	46.73	34.16	7.49	<b>54.12</b>	53.24
S_VAA-D_NE	40.79	8.14	59.58	44.91	39.11	8.15	57.19	47.39	33.44	7.38	53.91	54.02
S_VAA-D_NE+NN	41.78	8.28	59.60	43.80	39.46	8.26	57.03	46.48	34.21	7.49	53.69	52.91
S_VAA-D_NN	41.41	8.22	59.68	44.30	39.66	8.22	57.34	46.89	33.83	7.44	53.69	53.58

Table 4. BLEU,NIST, METEOR and TER scores of the different experimental conditions for NIST test sets MT04-NW, MT05 and MT08-NW\*<sup>4</sup>

<sup>4</sup> The gray highlighted cells indicate enhancement over Baseline. The Bold underlined score per column is the best score for that Testset/Metric. (Note: lower TER scores indicate better performance)

Zone integration is not helping (except non-significantly<sup>5</sup> for NEs on MT08-NW), this is due to the fact that marking MWEs as zones and enforcing decoder to respect these zones does not prevent the decoder from translating MWEs compositionally. While the decoder is not allowed to translate out of zone phrases unless it fully finishes translating the words in the zone, it is permissible to divide the zone into any combination of phrases and translate these phrases individually and in any order.

Following are the translation of our sample test sentence for selected conditions:

-Ref:

vm An gzw krdstAn AlErAq lm yEd mhmp shlp .

-Baseline:

gzw ErAqy krdstAn lm yEd shlA .

-S VAA:

gzw ErAqy krdstAn lys mhmp shlp .

-S NN:

gzw iraqis\_kurdistan lm yEd shlA .

-S VAA+NN:

gzw iraqis\_kurdistan lm tEd mhmp shlp .

-Z VAA+NN:

gzw ErAqy krdstAn lm yEd shlA .

## 7 Conclusion

Our study indicates that, at least for our language pair, different MWE types require different integration methods in the SMT pipeline where the more flexible an MWE is, the more the dynamic the integration needs to be. Therefore, for NE and NN, dynamic integration yields the best results. While for VAA, which tend to be more rigid, we gain the most from static integration.

Our results strongly suggest that explicit modeling for MWE and their various types definitely impact SMT performance positively. This is important since the number of MWE (VAA+NN+NE) tokens in the text only amounts to a total of 5.3% of the data, even though in terms of type ratio, MWEs (VAA+NN+NE) account for 46% of the types (indicating that we see a lot of variability in type but with very low frequency), yet we see gains of up to 0.82 absolute BLEU points (for A\_VAA-D\_NE+NN MT05). We anticipate such effects to be even

more pronounced in other more nuanced data sets such as blogs and broadcast conversations where the use of MWEs is pervasive compared to Newswire.

For future work, we plan to extend our matching algorithm to account for syntactically flexible MWEs by allowing gaps within MWE. We also plan to enhance feature engineering of the dynamic integration by assigning each MWE type a dedicated feature in the model. Finally we plan to extend our study to different language pairs and for MWEs in both source and target languages.

## Acknowledgments

We would like to thank the feedback provided by three anonymous reviewers.

This work was partially supported by the DARPA BOLT program.

## References

- Bai, M.H., You, J.M., Chen, K.J., and Chang, J. 2009. *Acquiring translation equivalences of multiword expressions by normalized correlation frequencies*. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 478–486.
- Banerjee, S., and Lavie, A. 2005. *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments*. In Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, Ann Arbor, MI, USA, 65–73.
- Bouamor, D., Semmar, N., Zweigenbaum, P. 2012. *Identifying bilingual Multi-Word Expressions for Statistical Machine Translation*. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey
- Carpuat, M., and Diab, M. 2010. *Task-based evaluation of multiword expressions: a pilot study in statistical machine translation*. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 242-245.
- Diab, M. 2009. *Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking*. MEDAR 2nd International Conference on Arabic Language Resources and Tools, April, Cairo, Egypt
- Diab, M., Hacıoglu, K., and Jurafsky, D. 2007. *Automatic Processing of Modern Standard Arabic Text*. In Arabic Computational Morphology: Knowledge-based and Empirical Methods, A. Soudi, A. Bosch and G. Neumann, Springer, The Netherlands, 159-180.

<sup>5</sup>Statistical significance tests use bootstrapping methods as detailed in (Zhang and Vogel, 2010)

- Doddington, G. 2002. *Automatic evaluation of MT quality using n-gram co-occurrence statistics*. In Proceedings of Human Language Technology Conference 2002, San Diego, CA, USA, 138–145.
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Finkel, J., Grenager, T., and Manning C. 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). Association for Computational Linguistics, Stroudsburg, PA, USA, 363–370.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantine, A., and Herbst, E. 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). Association for Computational Linguistics, Stroudsburg, PA, USA, 177–180.
- Lambert, P., and Banchs, R. 2005. *Data inferred multiword expressions for statistical machine translation*. In Proceedings of Machine Translation Summit X, Phuket, Thailand, 396–403.
- Och, F., 2003. *Minimum error rate training for statistical machine translation*. In Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, July
- Pal, S., Supin, K.N., Pavel, P., Sivaji, B., and Way, A. 2010. *Handling Named Entities and Compound Verbs in Phrase-Based Statistical Machine Translation*. In Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications (MWE 2010). Association for Computational Linguistics, Stroudsburg, PA, USA, 45–53.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.J. 2002. *BLEU: a method for automatic evaluation of machine translation*. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02). Association for Computational Linguistics, Stroudsburg, PA, USA, 311–318.
- Ren, Z., LÜ, Y., Cao, J., Liu, Q., and Huang, Y. 2009. *Improving statistical machine translation using domain bilingual multiword expressions*. In Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE '09). Association for Computational Linguistics, Stroudsburg, PA, USA, 47–54.
- Sag, I., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. 2002. *Multiword Expressions: A Pain in the Neck for NLP*. In Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '02), Alexander F. Gelbukh (Ed.). Springer-Verlag, London, UK, UK, 1–15.
- Schmid, H. 1994. *Probabilistic part-of-speech tagging using decision trees*. In Proceedings of International Conference on New Methods in Language Processing, Manchester, UK, 44–49.
- Snover, M., Dorr, B., Schwartz, R., Makhoul, J., and Micciulla, L. 2006. *A study of translation error rate with targeted human annotation*. In Proceedings of the Association for Machine Translation in the Americas Conference 2006, Boston, MA, USA, 223–231.
- Zhang, Y., Vogel, S. 2010. *Significance Tests of Automatic Machine Translation Evaluation Metrics*, In Machine Translation: Volume 24, Issue 1 (2010), Page 51–65.



# Uncertainty Detection for Natural Language Watermarking

György Szarvas<sup>1</sup> Iryna Gurevych<sup>1,2</sup>

(1) Ubiquitous Knowledge Processing Lab (UKP-TUDA)

Department of Computer Science, Technische Universität Darmstadt

(2) Ubiquitous Knowledge Processing Lab (UKP-DIPF)

German Institute for Educational Research and Educational Information

<http://www.ukp.tu-darmstadt.de>

## Abstract

In this paper we investigate the application of uncertainty detection to text watermarking, a problem where the aim is to produce individually identifiable copies of a source text via small manipulations to the text (e.g. synonym substitutions). As previous attempts showed, accurate paraphrasing is challenging in an open vocabulary setting, so we propose the use of the closed word class of uncertainty cues. We demonstrate that these words are promising for text watermarking as they can be accurately disambiguated (from the non-cue uses of the same words) and their substitution with other cues has marginal impact to the meaning of the text.

## 1 Introduction

The goal of digital watermarking is to hide digital information (a secret marker) in an audio stream, image or text file. These markers are by design not perceivable while listening, watching or reading the data, but can be read with a tailor-made algorithm and can be used to authenticate the data that carries it, or to identify its owner. We discuss the concept of a watermark and the process of embedding it in a media in more detail in Section 2. Text watermarking is a digital watermarking problem where the aim is to embed a secret message (a sequence of bits) in a *text* in order to make the actual text copy individually identifiable (Bergmair, 2007). That is, given a natural language source text (e.g. an ebook), the aim is to produce individual copies of the text by means of surface, syntactic or semantic manipulations. The individualized copies should preserve the readability and the meaning of the original text, i.e. the modifications should be undetectable to the readers.

The manipulation of the text can be performed

on the surface or the content level. Surface manipulation affects the visual appearance of the text, as in white space modulation. In contrast, syntactic reordering looks e.g. for conjunctions and reorders the connected words or phrases. Finally, semantic manipulation, such as *synonym substitution*, takes a target word and replaces it with a contextually plausible synonym. Consider the following examples:

He was *bright* and independent and proud.  
He was *bright* and independent and proud. (surface)  
He was independent and *bright* and proud. (syntactic)  
He was *clever* and independent and proud. (semantic)

In steganography, there is a natural tradeoff between capacity (the length of the secret message that can be embedded in a cover medium) and precision (how well the manipulations preserve readability and meaning). Text watermarking is a precision-oriented application as it requires very high transformation quality, and relatively low capacity (recall) is acceptable as the goal is to make a relatively small number of changes in a text. Surface manipulations are typically very easy to spot, so those are not realistic alternatives, while syntactic and semantic methods are equally viable. From a practical perspective, the most reliable method should be used to embed the secret information. E.g. enumeration reordering is a relatively robust method, provided we have a good parser at hand to detect the conjoined units. On the other hand, if there is a reference in the near context, it can be more reliable to employ synonym substitution:

He made offers to John and Mary. The latter *accepted*.  
He made offers to John and Mary. The latter *agreed*.

Here we investigate paraphrasing as a way to embed secret information in texts. An open vocabulary approach to synonym substitution could ensure rather high capacity. For example, Chang and Clark (2010b) suggest to identify one paraphrase position per sentence, thereby enabling the system to encode one bit per sentence, but they achieve

original bitmap	#	after embedding	#
1 1 1 0 0 0	1	1 1 1 0 0 0	1
1 1 1 1 0 1	1	1 1 1 0 0 1	0
1 0 1 0 0 1	1	1 0 1 1 0 1	0
1 1 0 0 0 1	1	1 1 0 0 0 1	1
1 0 0 0 0 1	0	1 0 0 0 0 1	0
0 0 0 0 1 1	0	0 0 0 0 1 1	0

Table 1: Example watermark.

this at the cost of relatively low precision (around 70% even when using the information about the correct word sense). In contrast, we propose to use a closed word class – i.e. uncertainty cues – for paraphrasing. Semantically uncertain propositions are common in natural language: they either directly express something as uncertain (epistemic modality); assert the speaker’s hypotheses, beliefs; express events that are unconfirmed or under investigation or conditional to other events, etc. These linguistic devices are lexical in nature, i.e. they are triggered by the use of specific keywords in the sentence, which we refer to as *uncertainty cues*. These words are good targets for paraphrasing since – when replaced by another uncertainty cue with similar properties (part of speech and meaning) – their substitution does not change the meaning of the sentence: the main proposition (“who does what, when and where”) remains the same and the proposition stays uncertain:

The legislative body *may* change.  
The legislative body *might* change.

That is, in our approach a classifier first detects uncertainty cues in a text. Then, those disambiguated cues that are both detected with high precision and have such paraphrases that are valid in all “uncertain” contexts can be replaced with their paraphrase. This manipulation affects the watermark bit contained in the passage and thus allows information encoding. The actual process is described in more detail in Section 2.1.

With this work, we show that a closed-class approach to generating paraphrases has desirable properties for watermarking: almost perfect substitution quality and acceptable embedding capacity. At the same time, we propose a novel application of uncertainty cue detection: paraphrasing of uncertainty cues for linguistic steganography.

## 2 Digital Watermarking

In this section we elaborate on what a watermark message is. In the simplest setup, we can define a watermark message as a sequence of  $k$  bits (0 or 1 values). Then the message then can take  $2^k$  different values, and can be used to identify the owner

of the medium, if each owner gets a copy of the data with a different digital message embedded in it. Take, for example, the problem of embedding a 6-bit message in a black-and-white bitmap image. In this case, we divide the bitmap to six equal-size parts and consider the parity of the sum of bits in a given part as the message bit. Table 1 shows a  $6 \times 6$  bitmap and the embedding of a 6-bit message (100100, one bit in each line). For a comprehensive overview of the different watermarking techniques, we refer the reader to Cox et al. (2008).

### 2.1 Natural Language Watermarking with Paraphrases

In order to encode a single bit in a larger block of text – like a paragraph or section – based on bit parity, we first assign a bit (0 or 1) to every word in the vocabulary, i.e. not just those that we plan to manipulate. Then, in each block of the source text, we sum the bits encoded by the original text and take its parity (*even* = 0, *odd* = 1) as the secret bit. In case our message should contain the other bit, we have to make exactly 1 synonym substitution, replacing a 0-word with a 1-word or vice versa to reverse the parity of the block, in order to embed the desired bit in the text. Reading the message only requires the same methodology to detect the blocks (sections) in the text and the initial word-to-bit mapping to calculate the parity of the blocks.

If we consider only a small set of words as candidates for substitution, this places a constraint on the system: it has to be ensured that the receiver can identify which blocks are used to actually encode information. This can be done, for example, if we use such blocks that contain at least one paraphrasable word after the potential manipulation: this way the reader can be sure the actual block encodes a bit (as there is capacity left in the block).

This simple parity-based message encoding approach offers straightforward ways to combine different embedding techniques, which is desirable: in order to combine conjunction reordering with the proposed paraphrasing method, one could use the (parity of the) number of changes in the conjoined list of items from their lexicographic order as the bit encoded in a conjunction, and can use that bit in the message encoding process. In our application scenario, we plan to implement several different text manipulation methods (and combine

them based on their confidence). Therefore the goal of this study is to propose an approach with high precision and coverage will be ensured by a range of syntactic methods and paraphrasing together.

### 3 Related Work

In this section we briefly introduce the previous work in *natural language watermarking* and in *uncertainty detection*.

The most prominent previous work in natural language watermarking (Atallah et al., 2003) focus on manipulating the sentence structure in order to realize meaning-preserving transformations of the text, which in turn can be used to embed information in the text. This approach either requires the presence of a robust syntactic (and semantic) parser to construct the representation which supports complex sentence-level transformations, or it has to be simplified to local transformations (e.g. conjunction modulation, as in the examples above) in order to ensure high precision without the requirement of deep linguistic analysis. Unfortunately, robust syntactic and semantic parsing of arbitrary texts is still challenging for natural language processing. This fact justifies the importance of more shallow models, such as synonym substitution, provided these approaches can ensure high precision and robust performance across domains. For a general and detailed overview of linguistic steganography, including methods other than paraphrasing, see for example Bennett (2004) and Topkara et al. (2005).

#### 3.1 Paraphrasing for Linguistic Steganography

As regards synonym substitution, the first studies made no use (Topkara et al., 2006) or just limited use (Bolshakov, 2005) of the context through collocation lists. While this approach offers a relatively high capacity, the transformations result in frequent semantic, or even grammatical, errors in the text, which is undesirable (Bennett, 2004).

Recently Chang and Clark (2010a,b) proposed the use of contextual paraphrases for linguistic steganography. This offers a higher perceived quality and is therefore more suited to text watermarking where quality is a crucial aspect. Chang and Clark (2010b) used the English Lexical Substitution data (McCarthy and Navigli, 2007) from SemEval 2007 for evaluation, WordNet as the

source of potential synonyms and n-gram frequencies for candidate ranking to experiment with paraphrasing for linguistic steganography. They introduced a graph coloring approach to embed information in a text through the substitution of words with their WordNet synonyms. They report an accuracy slightly above 70% for their paraphrasing technique and a potential capacity of around one bit per sentence.

Chang and Clark (2010a) used a paraphrase dictionary mined from parallel corpora using statistical machine translation (SMT) methods. The ranking of candidate paraphrases was also based on n-gram frequencies and for a set of 500 paraphrase examples they reported 100% precision (with very low, 4% recall) for substitution grammaticality. The possible change in meaning for otherwise grammatical replacements was not evaluated.

#### 3.2 Uncertainty Cue Detection

Another major field of work related to this study is the detection of uncertainty cues, which we propose to use for paraphrasing in Section 4. The first approaches to uncertainty detection were based on hand-crafted lexicons (Light et al., 2004; Saurí, 2008). In particular, ConText (Chapman et al., 2007) used lexicons and regular expressions not only to detect cues, but also to recognize contexts where a cue word does not imply uncertainty.

Supervised uncertainty cue detectors have also been developed using either token classification (Morante and Daelemans, 2009) or sequence labeling approaches (Tang et al., 2010). A good overview and comparison of different statistical approaches is given in Farkas et al. (2010). Szarvas et al. (2012) addressed uncertainty cue detection in a multi-domain setting, using surface-level, part-of-speech and chunk-level features and sequence labeling (CRF) models. They found that cue words can be accurately detected in texts with various topics and stylistic properties. We make use of the multidomain corpora presented in their study and evaluate a cross-domain cue detection model for text watermarking.

### 4 Uncertainty Cue Detection for Text Watermarking

In this section we experiment with uncertainty cue detection and paraphrasing, and study the potential of this approach for text watermarking.

## 4.1 Dataset, Experimental Setup and Evaluation Measures

We used here the dataset introduced by Szarvas et al. (2012). It consists of texts from three different domains (articles from Wikipedia, newspaper articles and biological scientific texts) and is annotated for uncertainty cues. The uncertainty cues in the corpora are marked only in contexts where they are used in an uncertain meaning, i.e. these texts can be used to train a shallow (*cue* vs. *non-cue* meaning) disambiguation model for these phrases. Here we aim to paraphrase the *cue* uses of the words to encode information via cue-to-cue paraphrasing. We train and test our models on separate parts of the corpora: for example, to assess the accuracy of cue detection in Wikipedia texts, we train the model only on newswire and scientific texts, and so on. This is a cross-domain evaluation setting and is therefore a good estimate of how the system would perform on further, yet different text types from our training datasets.

For evaluation, we use the overall precision of the recognized uncertainty cues, and we also measure the capacity that can be achieved by paraphrasing these words, i.e. how frequently one of these words that we use to encode information actually occurs in a text (the number of detected objects divided by the number of sentences processed). These two criteria – precision and capacity – measure how well the uncertainty detector would perform as a stego system. In addition, we also perform an error analysis of the top-ranked instances that received the highest posterior probability scores by the classifier. The highest-ranked instances are especially important as the underlying application would chose the highest-ranked instance in a larger block of text to actually implement a change to the text.

## 4.2 Uncertainty Detection Model

We implemented a cue recognition model similar to that described in Szarvas et al. (2012), using simple features that are robust across various text types. This is important, as we plan to use the model for text types that can be different from those in the training corpus, and for which NLP modules such as parsers might have questionable performance.

Conditional Random Fields were found to provide the best overall results in cue detection. However, the relative advantage of sequence taggers

corpus	# sent.	# cues	precision	capacity	F(cue)
Wiki	20756	3438	69.69%	16.56%	71.28%
news	3123	522	84.48%	16.71%	70.33%
sci	19473	3515	91.58%	18.05%	70.92%

Table 2: Summary of cue recognition results.

over simple token-based classifiers is more prominent for less frequent, long cue phrases<sup>1</sup>. Since in this study we concentrate on the more simple and frequent unigram cues (or fixed constructions, such as *not clear*), we use a Maximum Entropy (Maxent) classifier model (McCallum, 2002) in our experiments for cue detection and disambiguation.

In our classification setup, each token is a separate instance, described by a set of features collected from a window of size 2 around the token. That is, each feature can be present under 5 different names, according to their relative position, except for sentence beginning and ending tokens (out-of-sentence positions are discarded). We used the following features to describe each token: i) lexical features (word base forms and word bigrams); ii) 3 to 5 characters long word prefixes and suffixes; iii) word-surface-level features denoting whether the word is written in all uppercase, uppercase initial, or lowercase form, it is a punctuation mark or number; and iv) part of speech tags.

## 4.3 Results

The results of the cross-domain cue recognition experiments are summarized in Table 2. The columns indicate the total number number of sentences and recognized cues in the corpora, their precision and the capacity that can be achieved via cue paraphrasing. For comparison to previous works, we also provide the overall phrase-level F score for uncertainty cues. These numbers are slightly better than those reported by Szarvas et al. (2012) for cross-training with CRFs, probably due to the different settings (we used two domains for training, not just one).

As can be seen, the uncertainty cue recognizer is accurate even in a challenging cross-training setting: the precision is well above 80% for two out of three domains, and is around 70% for the most difficult Wikipedia dataset. This precision could realize a capacity of one bit per every six sentences, on average (capacity at or above 16%). In

<sup>1</sup>We performed an initial experiment using token-based and sequential models on our corpora and found no statistically significant difference in performance on unigram cues.

corpus	# sent.	# cues	precision	capacity
Wiki	20756	1869	89.46%	9.00%
news	3123	223	93.72%	7.14%
sci	19473	2688	98.95%	13.80%

Table 4: Summary of results with the 29 selected keywords.

order to use this cue recognizer as a watermarking method with the above-mentioned precision and capacity, we should provide a valid paraphrase for all of the 300 uncertainty cues found in the corpora. Doing that, a precision of 70% or more is promising in the light of the precision values below 50% for the first answers at SemEval 2007 for an all-words substitution task (Yuret, 2007), and of the fact that this precision stays around 70% even if the correct sense is picked in advance based on the human answers (Chang and Clark, 2010b).

On the other hand, as we argued in Section 2.1, it is desirable to improve precision at the cost of capacity. Thus, we filtered the uncertain vocabulary for such cues that are both frequent and accurate: we kept the cues that had a frequency of at least 10, with a precision above 80%. This left us 37 cues in total and this list was given to two annotators to provide paraphrases. The annotators were told to perform a web search for various contexts of the words and suggest paraphrases that are acceptable in all the words’ uncertainty-cue-uses and contexts. The two annotators agreed on a unique paraphrase which fits in all uncertain contexts of the target words for only 29 cues. These words with the proposed paraphrases and examples from the Wikipedia corpus are listed in Table 3. The columns indicate the selected cue words with their part of speech and the proposed paraphrase cue (or *XX* where we could not provide a suitable paraphrase). As can be seen, each cue is paraphrased with another cue with the same part of speech. Thus, their inflected forms can be generated based on the original words’ POS tags. For the remaining eight cues the annotators either did not find a proper substitute (e.g. *belief*) or found the word to have several uncertain readings which would require different paraphrases in different contexts (e.g. *expect* which can be rephrased as *wait*, *hope*, *count on*, ...).

Table 4 provides aggregate numbers for the selected 29 cue words. The columns indicate the total number of sentences in the corpora, the number of recognized instances of the 29 se-

lected cues and their precision and capacity. As can be seen, for these words the classifier yielded excellent precision scores, even with cross-domain training. In scientific and newswire texts, the model performs well above 90% precision, while in Wikipedia texts the precision is slightly lower.

As regards the capacity of the selected cues in the texts, on average we can find one instance of the selected cue words in every 7–14 sentences. The above precision and capacity scores can be realized in an actual watermarking method with the use of the paraphrases in Table 3. While this coverage seems lower than some other approaches (e.g. Chang and Clark (2010b) can achieve approximately one bit per sentence capacity), it is still acceptable for text watermarking of lengthy documents (such as ebooks), and as mentioned earlier, different methods can be combined to increase capacity. In the light of this, we consider our results especially promising, due to the remarkable precision scores, and the positive characteristic that these changes do not affect the main propositions in the sentences, i.e. meaning is well preserved.

Although direct comparisons are difficult to make, Chang and Clark (2010a) evaluated how accurately their model predicted a paraphrase to be grammatical, which is similar to our goal here. Their model achieved similar precision levels with similar or slightly lower potential capacity scores. Other previous approaches reported substantially lower precision (typically aiming to achieve high recall). These results suggest that our methodology, making a change in 7–14% of the sentences with a precision of 90–98% is a very competitive alternative for precision-oriented linguistic steganography.

#### 4.4 Error Analysis

The proposed method can embed information in a cover text with remarkably high precision, as the applied changes to the text are perfectly grammatical and do not affect the main aspects of the meaning of the text. Our error analysis also confirms this (see Appendix). We checked the 250 top-ranked classifications in the Wikipedia and scientific text corpora, and 223 classifications in the newswire texts (the total number of detected instances of the 29 selected cues). We checked the errors in the instances that obtained the highest posterior scores because in a larger block the sys-

WORD	POS	SUBST.	Example
accuse (of)	V	blame (for)	Certain corporations have been <b>accused</b> of paying news channels.
allege	V	claim	A friend of his <b>alleges</b> it detects ghosts.
allegedly	RB	reportedly	Britain was <b>allegedly</b> fighting for the freedom of Europe.
assume	V	hypothesize	It is <b>assumed</b> that women are not capable of inflicting such harm.
assumption	N	hypothesis	They respond that the <b>assumption</b> has long been that he worked from a sketch.
belief	N	XX	It was common <b>belief</b> that all species came to existence by divine creation.
believe	V	think	These were <b>believed</b> to be in the CA \$150,000 range.
determine	V	XX	... but ongoing studies have <b>yet to determine</b> to what degree.
expect	V	XX	He <b>expects</b> to be promoted to a grade 35 bureaucrat.
hypothesize	V	assume	It is <b>hypothesized</b> that most of these chemicals help.
hypothesis	N	assumption	There is some evidence to support the <b>hypothesis</b> that they undergo fission.
idea	N	XX	The <b>idea</b> that it constitutes an edifice was publicized by Osmanagic.
imply	V	denote	It is <b>implied</b> to be the center of the Dust Factory.
indicate	V	suggest	It <b>indicates</b> that there are good opportunities for skilled people.
likely	JJ	probable	It is <b>likely</b> that they were instructed by their grandmother M. V. van Aelst.
likely	RB	probably	The camps will <b>likely</b> never reopen as their locations posed lightning risks.
may	MD	might	The legislative body <b>may</b> change or repeal any prior legislative acts.
might	MD	may	Other instruments that <b>might</b> be connected are air data computers.
not clear	JJ	unclear	How the plant arrived on the island is <b>not clear</b> .
perhaps	RB	maybe	His work was <b>perhaps</b> known to Islamic mathematicians.
possibility	NN	potential	However the <b>possibility</b> of merging University Park with Downtown LA remains years away.
possibly	RB	potentially	It is <b>possibly</b> a close relative to the dwarf flannelbush species.
presumably	RB	supposedly	Wellstone was <b>presumably</b> worried about money from rich individuals.
probably	RB	likely	He was <b>probably</b> better known for his antics than his pitching talent.
regard	V	XX	Shea Fahy is <b>regarded</b> as one of the heroes of the team.
seem	V	appear	It <b>seems</b> that Kev takes the opportunity to ...
seemingly	RB	apparently	Pelham was <b>seemingly</b> intimate with John Smibert.
speculate	V	assume	Some people <b>speculate</b> that these compounds are linked to health concerns.
suggest	V	indicate	It <b>suggests</b> that those few cases have their needs already met.
suppose	V	assume	The "arms" of the bow are <b>supposed</b> to cross each other.
suspect	V	XX	The diagnosis is often <b>suspected</b> on the basis of tests.
think	V	XX	Most people did not think that the Rams belonged on the same field with the Steelers.
thought	V	XX	Sleep is <b>thought</b> to improve the consolidation of information.
unclear	JJ	not clear	It is <b>unclear</b> whether it was House or Wilson.
unlikely	JJ	not likely	Historians are <b>unlikely</b> to fully understand which species were used in medicine.
view (that)	N	opinion (that)	... that undermined their capacity to accept the <b>view</b> that socialist incentives would not work
whether	IN	if	... or <b>whether</b> his paintings were purchased by Italians.

Table 3: Uncertain paraphrase dictionary with examples from the Wikipedia corpus.

tem would select the most confident position to perform a substitution, so precision at top ranks is the most important. We found 20 false positive classifications in these 723 sentences, attributed to eight different keywords. This is above 97% precision at top ranks, the errors are detailed in the appendix, together with example sentences. As can be seen, many of these misclassifications actually do not do any major harm to the meaning of the text: some of these high-ranked examples are actually replaceable with the proposed cover words even in a non-cue usage (this is the reason for their high posterior score).

## Conclusion and Future Work

In this paper we proposed uncertainty cue detection and the paraphrasing of uncertainty cues as a new approach to linguistic steganography. We experimented with texts from three different domains using cue detection models trained on out-of-domain texts in order to simulate a realistic application scenario. We found that uncertainty cues are capable of embedding a 1-bit message in a block of text of around 10–13 sentences on average. Although this capacity is limited, in turn the use of uncertainty cues offers nearly perfect

precision, i.e. the manipulated texts are grammatical and preserve the original text’s meaning. As in text watermarking the goal is to embed a relatively short message in potentially large texts, but with high precision (quality), the paraphrasing of uncertainty cues is a promising alternative.

Arguably, the ideal setting from an application perspective would be an open-vocabulary substitution system, but such an approach suffers from significant limitations stemming from the difficulty of paraphrasing in a general setting. An alternative could be to use a larger set of frequent words in a lexicalized approach for lexical substitution which might offer higher capacity with comparable precision Biemann and Nygaard (2010).

On the other hand, we think it is a viable approach to target the extra-propositional aspects of meaning (such as uncertainty proposed here) for lexical substitution in linguistic steganography. This – by definition – leaves the main proposition of the sentence (*who does what, when and where*) untouched, ensuring high transparency. To this end, we also plan to extend the capacity of this approach via paraphrasing other word classes such as opinion expressions (e.g. *great X* for *excellent X*, *awful X* for *terrible X*).

## Appendix

WORD	#	SUBST.
suggest	2	indicate
may	6	might
might	2	may
likely	1	probable
believe	2	think
assume	3	hypothesize
indicate	3	suggest
possibility	1	potential

Table 5: Examples for the 21 errors in the top 250 (wiki, sci) and 223 (news) examples.

### Example Errors

- Churchill wrote to him **suggesting** that he would sign his own works "Winston S. Churchill".
- He **may** be an idiot savant, but he's got great hair.
- It's fairly intense as you **might** well imagine.
- One big question now is the **likely** role of Mr. Fournier's allies.
- Nobody **believe** this any more.
- Cilcior will also **assume** 22 million of Hunter's existing debt.
- Kellogg **indicated** that it has room to grow without adding facilities.
- The second **possibility** would be to start a fight with Israel.

### Acknowledgment

This work was supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program (I/82806), and by the German Ministry of Education and Research under grant *SiDiM* (grant No. 01IS10054G).

### References

- M. J. Atallah, V. Raskin, C. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg. 2003. Natural language watermarking and tamperproofing. In *Proc. of 5th Workshop on Information Hiding*, pages 196–212.
- K. Bennett. 2004. Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. Technical report, Purdue University.
- R. Bergmair. 2007. A comprehensive bibliography of linguistic steganography. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*.
- C. Biemann and V. Nygaard. 2010. Crowdsourcing wordnet. In *Proceedings of the 5th Global WordNet conference*.
- I. Bolshakov. 2005. A method of linguistic steganography based on collocationally-verified synonymy. In *Information Hiding*, volume 3200 of *LNCS*, pages 607–614.
- C-Y. Chang and S. Clark. 2010a. Linguistic steganography using automatically generated paraphrases. In *Proceedings of NAACL 2010*, pages 591–599.
- C-Y. Chang and S. Clark. 2010b. Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In *Proceedings of the EMNLP 2010*, pages 1194–1203.
- W. W. Chapman, D. Chu, and J. N. Dowling. 2007. ConText: An algorithm for identifying contextual features from clinical text. In *Proceedings of BioNLP 2007*, pages 81–88.
- I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. 2008. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., 2 edition.
- R. Farkas, V. Vincze, Gy. Móra, J. Csirik, and Gy. Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of CoNLL: Shared Task*, pages 1–12.
- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA.
- H. Kilicoglu and S. Bergler. 2008. Recognizing speculative language in biomedical research articles: A linguistically motivated perspective. In *Proceedings of the BioNLP Workshop*, pages 46–53.
- M. Light, X. Y. Qiu, and P. Srinivasan. 2004. The language of bioscience: Facts, speculations, and statements in between. In *Proceedings of Biolink 2004 Ws.*, pages 17–24.
- A. K. McCallum. 2002. *MALLET: A Machine Learning for Language Toolkit*. <http://mallet.cs.umass.edu>.
- D. McCarthy and R. Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of SemEval-2007*, pages 48–53.
- B. Medlock and T. Briscoe. 2007. Weakly Supervised Learning for Hedge Classification in Scientific Literature. In *Proceedings of ACL*, pages 992–999.
- R. Morante and W. Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the BioNLP 2009 Workshop*, pages 28–36.
- R. Saurí. 2008. *A Factuality Profiler for Eventualities in Text*. Ph.D. thesis, Brandeis University, Waltham, MA.
- Gy. Szarvas, V. Vincze, R. Farkas, Gy. Móra, and I. Gurevych. 2012. Cross-Genre and Cross-Domain Detection of Semantic Uncertainty. *Computational Linguistics*, 38(2):335–367.
- B. Tang, X. Wang, X. Wang, B. Yuan, and S. Fan. 2010. A cascade method for detecting hedges and their scope in natural language text. In *Proceedings of CoNLL-2010: Shared Task*, pages 13–17.
- M. Topkara, C. M. Taskiran, and E. J. Delp. 2005. Natural language watermarking. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 441–452.
- U. Topkara, M. Topkara, and M. J. Atallah. 2006. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th workshop on Multimedia and security*, pages 164–174.
- D. Yuret. 2007. Ku: Word sense disambiguation by substitution. In *Proceedings of SemEval-2007*, pages 207–214.

# KySS 1.0: a Framework for Automatic Evaluation of Chinese Input Method Engines\*

Zhongye Jia and Hai Zhao<sup>†</sup>

MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems,  
Center for Brain-Like Computing and Machine Intelligence  
Department of Computer Science and Engineering, Shanghai Jiao Tong University  
800 Dongchuan Road, Shanghai 200240, China  
jia.zhongye@gmail.com, zhaohai@cs.sjtu.edu.cn

## Abstract

Chinese Input Method Engine (IME) plays an important role in Chinese language processing. However, it has been subjected to lacking a proper evaluation metric for a long time. The natural metric for IME is user experience, which is a rather vague goal for research purpose. We propose a novel approach of quantifying user experience by using keystroke count and then correspondingly develop a framework of IME evaluation, which is fast and accurate. With the underlying linguistic background, the proposed evaluation framework can properly model the user behavior as Chinese is input through English keyboard. It is helpful to point out a way to improve the current Chinese IME performance.<sup>1</sup>

## 1 Introduction

Chinese IME is a software solution that enables user to input Chinese into computer with a reasonable size keyboard, by mapping Chinese characters into English letter combinations. Nowadays the majority of Chinese IMEs are pinyin based. Pinyin is originally designed as the phonetic symbol of a Chinese character, using Latin letters as its syllable notation. For example, the pinyin of the Chinese character “爱”(love) is “ài”.<sup>2</sup> There are only less

than 500 pinyin syllables in standard modern Chinese, compared with over 6,000 commonly used Chinese characters, which leads to serious ambiguity for pinyin-to-character mapping. Other IMEs using various mapping scheme more or less share this same ambiguity problem, although some of them such as five-stroke IME<sup>3</sup>, may have lower ambiguity but they are very difficult to learn. So at present pinyin IMEs are the most popular, we focus on them in this paper. Modern IMEs are *sentence-based* (Chen and Lee, 2000) to reduce the ambiguity, which means that the IME generates a character sequence upon a sequence of English letter inputs, e.g. pinyin syllables. This is a non-typical sequence labeling task, as English alphabet input is the original sequence and Chinese characters are target labels. IMEs usually utilize beam search incorporated with language model initialized by (Chen and Lee, 2000), then later extended by (Liu and Wang, 2002), (Lee, 2003), and (Zhang, 2007). There are also various methods using conventional sequence labeling techniques such as: support vector machine (Jiang et al., 2007), maximum entropy model (Wang et al., 2006), conditional random field model (Li et al., 2009) and machine translation (Yang et al., 2012a), etc. IME also attracts attention from major software and internet corporations, Microsoft<sup>4</sup>, Google<sup>5</sup> and many others developed their own IME products.

Note that “sentence” from an IME’s viewpoint is not the linguistic “sentence”. It is actually the *Max Input Unit* (MIU), the longest consecutive Chinese character sequence inside a sentence. For example, the sentence “第51届ACL年会即将召开” (The 51st annual meeting of ACL will open soon)

\*This work was partially supported by the National Natural Science Foundation of China (Grant No.60903119, Grant No.61170114, and Grant No.61272248), and the National Basic Research Program of China (Grant No.2009CB320901 and Grant No.2013CB329401).

<sup>†</sup>Corresponding author

<sup>1</sup>KySS is the abbreviation for: **KeyStroke Score**

<sup>2</sup>Chinese is a language with tone, but for nearly all pinyin IMEs, tone marks are omitted since it is inconvenient to input.

<sup>3</sup><http://www.wangma.com.cn/>

<sup>4</sup><http://bing.msn.cn/pinyin/>

<sup>5</sup><http://www.google.com/intl/zh-CN/ime/pinyin/>



has 3 MIUs: “第”, “届” and “年会即将召开”. For more precise expression, we will use “MIU” or “character sequence” throughout this paper as far as possible, instead of “sentence”.

Chinese is used by the largest population in the world, and IME is necessary for all Chinese computer users. However according to our best knowledge there are no comprehensive and quantified metrics for evaluating its performance. The existing evaluation metrics for IME (Chen and Lee, 2000; Yang et al., 2012a) only focus on the character sequence generation, by adopting typical sequence labeling measurements such as character-wise precision and recall, sequence error rate and so on. However IME is an application deeply involving human-computer interaction, user behavior ought to be taken into account for IME evaluation (Zheng et al., 2011). Rather than merely judging the performance of character sequence generation, a good IME evaluation system should properly model the user behavior.

Compared with other typical evaluation systems for NLP tasks, such as the well known machine translation evaluation system BLEU (Papineni et al., 2002) and the summarization evaluation system ROUGE (Lin, 2004), they measure the closeness of machine translation/summarization and human results, using some  $n$ -gram co-occurrence statistics (Lin and Hovy, 2003), while our IME evaluation framework tries to quantify user experience during inputting Chinese. Existing keystroke based methods mostly focus on the keystroke duration and frequency pattern for biometric authenticate system security (Giot et al., 2009; El-Abed et al., 2012), instead of user experience modeling.

## 2 The Evaluation System

### 2.1 The User-IME Interaction

As introduced in Section 1, nearly all IMEs nowadays are “sentence based”. However, IME does not always succeed to exactly generate the entire expected character sequence for an input letter sequence. Thus IMEs output a list of all possible *candidate* character sequences corresponding to the input sequence.

Suppose that the input pinyin sequence is  $S_1S_2\dots S_n$  where  $S_i$  is a pinyin syllable with index  $i$  and  $n$  is number of pinyin syllables. The *best character sequence* has the same length as the input pinyin sequence, and is always in the first position of the list. Other output character sequences in

the candidate list is given according to  $S_1S_2\dots S_{n'}$  where  $n'$  is usually less than  $n$ . As  $n$  is usually large, it is unlikely that the best character sequence is completely correct, but for those shorter character sequences in the candidate list, user can always find that one of them may partially match his/her input target. If user has selected a candidate from the list to partially complete the input, for example, the character sequence for  $S_1S_2\dots S_j$  has been determined by this selection, then IME will dynamically output a new list of character sequences for the rest pinyin sequence  $S_{j+1}S_{j+2}\dots S_n$ . User can continue to make the selection from the list until the desired input is accomplished. For a candidate, we define its position in candidate list as its *rank*,  $r$ . The best character sequence has  $r = 0$ . The candidate at the  $j$ -th position in the list has  $r = j - 1$ .

There are often dozens of candidates for a pinyin sequence, as the space of input interface is limited, candidates have to be shown in multiple pages. A *page* is part of the candidate list. User can make the choice from the page by pressing candidate ID  $1, 2, \dots, m$  in the current page. If the expected word/character does not occur in the current page, user has to press a “next-page”<sup>6</sup> key to see more candidates.

Suppose that one wants to input the MIU “年会即将召开”(The annual meeting will open soon) by typing the pinyin sequence “nian hui ji jiang zhao kai”, a typical IME prompt window is like Figure 1a.

As shown in Figure 1a, the best character sequence is not completely correct for the input purpose. User has to pick up the 2nd candidate “年会”(annual meeting). Then the IME will update the window as Figure 1b. User can select the 3rd candidate “即将”(soon). The IME will update for the rest as shown in Figure 1c. The first candidate exactly completes the expected character sequence.

In the real world, user behavior is very difficult to predict. In order to alleviate the difficulty of user behavior modeling, an abstract assumption for user-IME interaction is proposed: all user input actions are only restricted to pinyin sequence input, candidate ID selection and page turning. In this manner, user inputs a sequence of pinyin then make a choice from the candidate list in the current page given by the IME, if the desired character se-

<sup>6</sup>Usually this “next-page” key is not the PageDown key, but mapped to some more convenient keys such as “+” or “.”.

nian hui ji jiang zhao kai	年会 ji jiang zhao kai	年会即将 zhao kai
1.年会激将召开	1.激将召开	1.召开
2.年会	2.激将	2.找
3.年	3.即将	3.赵
4.念	4.即	4.照
5.粘	5.及	5.招

(a) The first IME window      (b) The second IME window      (c) The third IME window

Figure 1: IME windows for inputting an MIU

quence is not presented in the current page, then user has to press “next-page” and goes on until the target is met.

Under this proposed user behavior model, user input is divided in to two parts: 1. a sequence of alphabet keys for pinyin and 2. a selection action sequence of “next-page” and candidate ID keys. The first part of inputting alphabet keys for different IMEs will be always the same, thus it can be ignored for evaluation metric. The second part would be the essential difference among different IMEs. We define the sequence of ranks of candidates as *Rank Sequence*, For an ideal rank sequence, it is always just  $\{0\}$ . For our previous example in this section, the actual rank sequence is  $\{1, 2, 0\}$ .

## 2.2 Evaluating IME

To make use of the rank sequence, i.e. keystroke count, as the metric to evaluate an IME, we define a few terms as follows:

- $\mathcal{L}$ : It is the length of MIU in characters.
- $\mathcal{P}$ : It is the length of rank sequence. It measures how many parts the MIU is split into to accomplish the input. In the previous example,  $\mathcal{P} = 3$  for “年会”, “即将” and “召开”. In the ideal situation,  $\mathcal{P}$  for any MIU is 1 since the exactly expected character sequence is rank at the top.
- $\mathcal{R}$ : It is the sum of all the elements in the rank sequence, i.e. the sum of ranks of candidates for each part. In our previous example,  $\mathcal{R} = 1 + 2 + 0 = 3$ . For the ideal situation,  $\mathcal{R}$  is always 0.
- $\mathcal{R}_W$ : It is the total keystroke without alphabet keys, by using the weighted sum of rank sequence:

$$\mathcal{R}_W = \sum_{i=1}^{k-1} \omega(r_i) + 1. \quad (1)$$

The weight function  $\omega(\cdot)$  reflects the cost of pressing “next-page” and candidate ID keys.

In the rest of this paper we will assume there are 5 candidates on each page which is the default setting for most existing IMEs. We also assume the keystroke cost of pressing numeric keys for candidate ID is 1 and the keystroke cost of pressing “next-page” is also 1, then the weight function is

$$\omega(r) = \lfloor \frac{r}{5} \rfloor + 1.$$

For example the 1st candidate on 3rd page has  $\mathcal{R}_W = 1 \times 2 + 1 = 3$ .  $\mathcal{R}_W$  measures how many keys the user has to press i.e. how much effort the user has to make to accomplish inputting an MIU.

- **KySS**: It is the final evaluation score of an IME. Consider an evaluation corpus  $\mathbb{C}$  with MIUs  $\{m_1, m_2, \dots, m_c\}$ , for a certain IME, the  $i$ -th MIU  $m_i$  has  $\mathcal{R}_W(i)$ , then the KySS score for the actual IME on the corpus is defined as the ratio between the total  $\mathcal{R}_W$  for the ideal IME and the total  $\mathcal{R}_W$  given by the actual IME:

$$\text{KySS} = \frac{\sum_{m_i \in \mathbb{C}} \mathcal{R}_W^{\text{ideal}}(i)}{\sum_{m_i \in \mathbb{C}} \mathcal{R}_W(i)} \quad (2)$$

$$= \frac{\|\mathbb{C}\|}{\sum_{m_i \in \mathbb{C}} \mathcal{R}_W(i)}. \quad (3)$$

For an ideal IME, we have  $\text{KySS} = 1$ . For actual IMEs,  $0 < \text{KySS} < 1$ . An IME with higher KySS is supposed to perform better.

## 3 Analysis on IMEs

### 3.1 Corpus

To build a corpus for evaluation, we extract 100,000 sentences from *China Daily* corpus<sup>7</sup>. We annotate it with pinyin sequence using the method in (Yang et al., 2012b). The corpus contains over 4 million characters, 87.6% of which are Chinese

<sup>7</sup>[http://www.icl.pku.edu.cn/icl\\_res/default\\_en.asp](http://www.icl.pku.edu.cn/icl_res/default_en.asp)

characters, other 12.4% are foreign letters, digits and punctuation. At last, a corpus with over 420,000 MIUs is built.

The IME for evaluation is sunpinyin<sup>8</sup> which is the state-of-the-art open-source Chinese pinyin IME on Linux developed by the former *Sun Microsystems, Inc.*

The distribution of  $\mathcal{L}$  for the evaluation corpus is shown in Figure 2. It can be seen that the length of

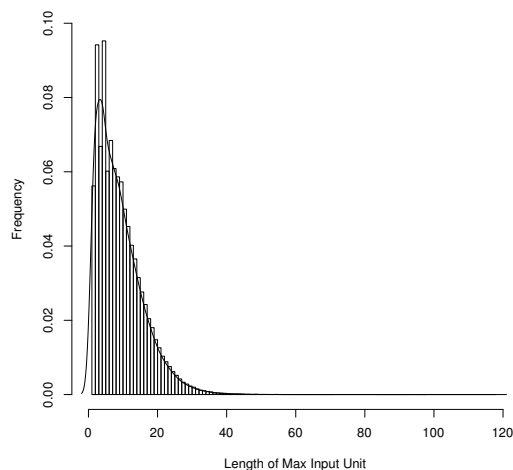


Figure 2: Distribution of  $\mathcal{L}$

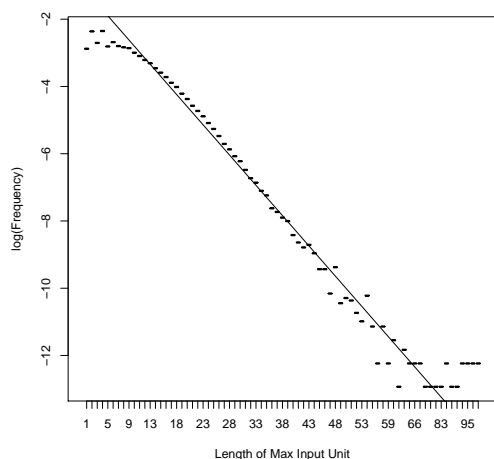


Figure 3: Linear fit on log-frequency over  $\mathcal{L}$

MIUs roughly follows an exponential function distribution in statistics. A linear regression on log-frequency is made and it fits the actual data well. The result is shown in Figure 3.

<sup>8</sup><http://code.google.com/p/sunpinyin/>

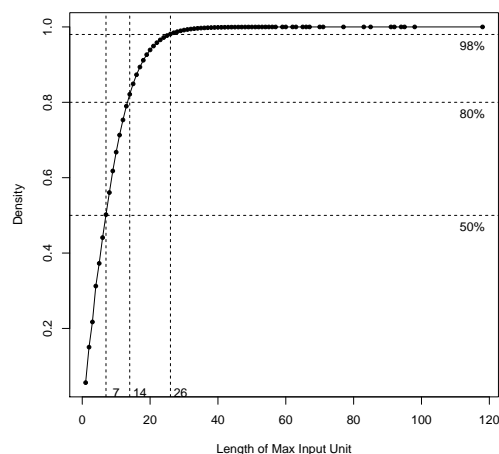


Figure 4: Cumulative distribution of  $\mathcal{L}$

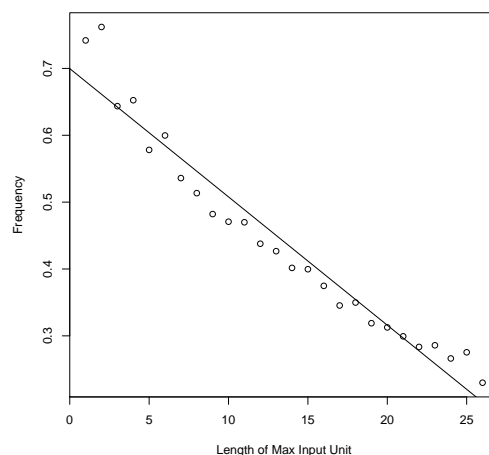


Figure 5: Rate of successful MIU generation

Although there are very long MIUs in the corpus, it can be observed from the cumulative distribution of  $\mathcal{L}$  in Figure 4 that most MIUs are short ones.

Paying so much attention to the length statistics of MIUs is necessary, because the core task of IME, character sequence generation, is heavily affected by the sequence length. The rate of successful character sequence generation decays linearly with the increment of  $\mathcal{L}$  as shown in Figure 5.

It also can be seen from Figure 4 that nearly 50% of MIUs are between 7 and 26 characters long, but as shown in Figure 5 they suffer from the rate of successful character sequence generation less than 0.5, which will be our major concern later.

### 3.2 Position Matters: How Character Sequence Generation Fails

From our empirical observation over those cases in which IME fails to generate the completely correct MIU, we found something interesting. The histograms of  $\mathcal{P}$  and  $\mathcal{R}$  of MIUs with  $\mathcal{L} = 10$  are shown in Figure 6, note that only unsuccessful results are plotted in the figure.

The interesting point is that a peak appears at  $\mathcal{L}/2$  in the histograms, as is shown in dotted lines in Figure 6. All the cases that we observed with length ranging from 7 to 26 demonstrate this “error peak” property.

We extract MIUs, IME generated best character sequences and rank sequences of those cases. The underlying reason is discovered after a manual inspection: in those best character sequences, the last error is at the rear of MIU. In such cases user has to select candidate words one by one until the end of the MIU.

For example, while inputting the MIU “大理 航空是个年轻的航空”(Dali airport is a new airport), the best character sequence generated by IME is “大力航展是个年轻的航展”(STRONG air show is a new air show). Its  $\mathcal{L} = 11$ . The selected candidates are: “大理”(Dali), “航空”(airport), “是个”(is), “年轻”(new), “的”(meaningless function word), and “航空”; so  $\mathcal{P} = 6$ . And the rank sequence is  $\{2, 1, 1, 1, 1, 1\}$  so  $\mathcal{R} = 7$ . So when errors occur at the rear of MIU, we have  $\mathcal{P} \approx \mathcal{N}_W$ , where  $\mathcal{N}_W$  is the number of words in MIU. For most of the real circumstances, those correctly generated words in the front part of MIU achieve a very high rank  $r$ , often the highest among candidates, i.e.,  $r = 1$  so  $\mathcal{R} \approx \mathcal{P} \approx \mathcal{N}_W$ . In sunpinyin, candidates except for the best character sequence are words queried from an internal dictionary. An important linguistic issue is that average length of Chinese words is about 1.8 characters, nearly two character long (Zhao et al., 2006). At last we obtain  $\mathcal{N}_W \approx \mathcal{L}/2$ , which explains the “error peak” at the middle position.

### 3.3 Improving IMEs with Evaluation Results

We propose a method to make use of the “error peak” for better IME performance. Among all those MIUs with the “error peak” problem, we can see an annoying situation is that all words except for the last few ones are successfully generated. In order to correct one or two words, user has to select word by word all the way through the character se-

quence. To avoid unnecessary selection, the IME can cut the front part of the generated character sequence and make it an independent candidate so that user can accomplish inputting as many words in one selection as possible. We suggest five cutting policies as the following:

- **Weakest**: Cut at the position where the conditional probability is smallest.
- **Mean, LtoR**: Cut at the position where the probability is lower than the mean of conditional probabilities of the sequence, scanning from left to right.
- **Mean, RtoL**: Similar as previous but scanning from right to left.
- **Halfway**: Cut at exactly at the middle position.
- **Fixed**: Cut a fixed length sequence. We experiment on length from 1 to 10 and found the best length is 3.

The KySS for each policy is shown in Table 1. The baseline is the original algorithm used by sun-

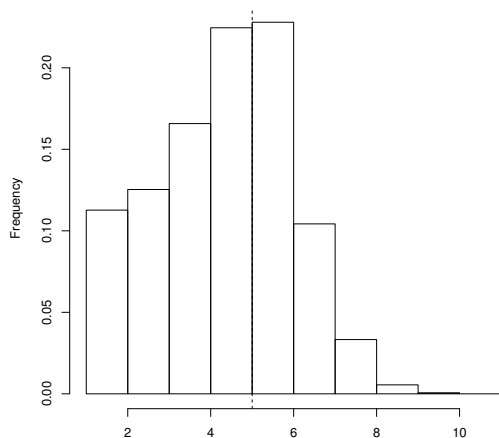
Policy	KySS
Baseline	27.67%
Weakest	29.41%
Mean, LtoR	29.21%
Mean, RtoL	29.06%
Halfway	30.71%
Fixed-3	31.30%

Table 1: KySS of each policy

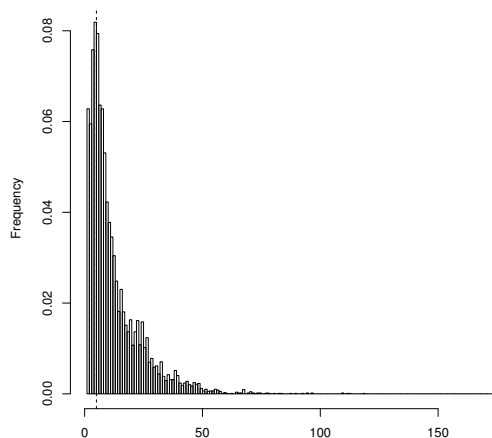
pinyin without any cutting. To our surprise, the dummy **Fixed-3** and **Halfway** cutting performs best with a more than 10% boost over the baseline due to its robustness. The **Fixed-3** and **Halfway** policy can always stably cut out a reasonable length of MIU. The problem with **Mean, RtoL** is that it is likely to cut at the end of MIU thus the head that it cuts out may still include errors. On the contrast, the **Mean, LtoR** policy tends to cut out too few words. And the **Weakest** policy is easy to fail because it often cuts a sequence with a low conditional probability but actually being correct.

## 4 Conclusion and Future Work

In this paper, a novel evaluation framework for Chinese IME, KySS, is proposed by effectively modeling user behavior during Chinese input. This evaluation framework aims to fast and accurately evaluate various IMEs from the view of user experience. It uses keystroke count as core metric.



(a) Histogram of  $\mathcal{P}$



(b) Histogram of  $\mathcal{R}$

Figure 6: Histogram of  $\mathcal{P}$  and  $\mathcal{R}$ , with  $\mathcal{L} = 10$

With the help of the framework we preliminarily propose a sequence cutting strategy to enhance the current IME.

The real world IME and user behavior can be very complicated. In this paper, we make a simplified assumption that all user input is correct. Unfortunately it may contain typos. And the IME may have prediction feature, i.e. generating character sequence longer than the input pinyin sequence. We may include those in our future work.

## References

- Zheng Chen and Kai-Fu Lee. 2000. A New Statistical Approach To Chinese Pinyin Input. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 241–247, Hong Kong, October.
- Mohamad El-Abed, Patrick Lacharme, and Christophe Rosenberger. 2012. Security evabio: An analysis tool for the security evaluation of biometric authentication systems. In *Biometrics (ICB), 2012 5th IAPR International Conference on*, pages 460–465. IEEE.
- Romain Giot, Mohamad El-Abed, and Christophe Rosenberger. 2009. Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–6. IEEE.
- W. Jiang, Y. Guan, XL Wang, and BQ Liu. 2007. Pinyin-to-character conversion model based on support vector machines. *Journal of Chinese information processing*, 21(2):100–105.
- Y.S. Lee. 2003. Task adaptation in stochastic language model for chinese homophone disambiguation. *ACM Transactions on Asian Language Information Processing*, 2(1):49–62.
- L. Li, X. Wang, X.L. Wang, and Y.B. Yu. 2009. A conditional random fields approach to chinese pinyin-to-character conversion. *Journal of Communication and Computer*, 6(4):25–31.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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.
- B.Q. Liu and X.L. Wang. 2002. An approach to machine learning of chinese pinyin-to-character conversion for small-memory application. In *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, volume 3, pages 1287–1291. IEEE.
- 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.

- X. Wang, L. Li, L. Yao, and W. Anwar. 2006. A maximum entropy approach to chinese pin yin-to-character conversion. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 4, pages 2956–2959. IEEE.
- S. Yang, H. Zhao, and B. Lu. 2012a. A machine translation approach for chinese whole-sentence pinyin-to-character conversion. In *Pacific Asia Conference on Language Information and Computation*, pages 367–376, Bali, Indonesia, November.
- S. Yang, H. Zhao, X. Wang, and B. Lu. 2012b. Spell checking for chinese. In *International Conference on Language Resources and Evaluation*, pages 730–736, Istanbul, Turkey, May.
- S. Zhang. 2007. Solving the pinyin-to-chinese-character conversion problem based on hybrid word lattice. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, 30(7):1145.
- Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of PACLIC*, volume 20, pages 87–94.
- Yabin Zheng, Lixing Xie, Zhiyuan Liu, Maosong Sun, Yang Zhang, and Liyun Ru. 2011. Why Press Backspace? Understanding User Input Behaviors in Chinese Pinyin Input Method. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 485–490, Portland, Oregon, USA, June. Association for Computational Linguistics.

# Automatic Extraction of Social Networks from Literary Text: A Case Study on *Alice in Wonderland*

**Apoorv Agarwal**  
Dept. of Computer Science  
Columbia University  
New York, NY, U.S.A.  
apoorv@cs.columbia.edu

**Anup Kotalwar**  
Microsoft, Inc.  
Redmond, WA, U.S.A.  
ankotalw@microsoft.com

**Owen Rambow**  
CCLS  
Columbia University  
New York, NY, U.S.A.  
rambow@ccls.columbia.edu

## Abstract

In this paper we present results for two tasks: *social event detection* and *social network extraction* from a literary text, *Alice in Wonderland*. For the first task, our system trained on a news corpus using tree kernels and support vector machines beats the baseline systems by a statistically significant margin. Using this system we extract a social network from *Alice in Wonderland*. We show that while we achieve an F-measure of about 61% on social event detection, our extracted un-weighted network is not statistically distinguishable from the un-weighted gold network according to popularly used network measures.

## 1 Introduction

Social network analysis affects a wide range of academic disciplines and practical applications: psychology (Seidman, 1985; Koehly and Shivy, 1998), anthropology (Sanjek, 1974; Johnson, 1994; Hage and Harary, 1983), political science (Knoke, 1990; Brandes et al., 2001), literary theory (Moretti, 2005), management (Tichy et al., 1979; Cross et al., 2001; Borgatti and Cross, 2003), and crime prevention and intelligence (Sparrow, 1991). In the past, social networks were constructed through interviews, surveys and experiments. With the advent of the internet and online social networks, researchers have started constructing networks using meta-data that reflects interactions, such as self-declared friendship linkages, sender-receiver email linkages, comments on a common blog-post, etc. However, these methodologies of creating social networks ignore a vastly rich network expressed in the unstructured text of such sources. Moreover, many rich sources of social networks remain

untouched simply because there is no meta-data associated with them (literary texts, news stories, historical texts). There have been recent efforts to extract social networks from text by mining interactions between people expressed linguistically in unstructured text (Elson et al., 2010; He et al., 2013). However, these approaches are restricted to extracting interactions signaled by quoted speech.

In this paper, we present results for extracting a social network from *Alice in Wonderland* that is not restricted to interactions signaled by quoted speech. We define a social network for a fictional text as follows: nodes are characters and links are *social events*. Two nodes in the network are connected if the characters engage in a social event. We introduced the notion of *social events* in our previous work (Agarwal et al., 2010), in which we presented our annotation scheme for annotating social events on the Automatic Content Extraction (ACE-2005) corpus. We presented a preliminary system for social event detection and classification in Agarwal and Rambow (2010). This system was trained and tested only on the ACE-2005 corpus. A priori, it is unclear if a system trained on a news corpus will be able to extract a high quality social network from a text from a very different genre (literary fiction). There are many syntactic and lexical differences between these genres. For example, news corpora have almost no questions, very little dialog presented as direct speech, and very little use of the first and second person pronouns. The vocabulary in literature can also be very different (relating to, say, whaling, passion, or teenage angst rather than current events). In this paper, we make two novel contributions. First, in an intrinsic evaluation, we show that our system without any domain (or genre) adaptation performs reasonably well on a new genre. Second, in an extrinsic evaluation, we show that the social network that our system extracts is not statistically distinguishable from the underlying gold network

in terms of various standard and popularly used network analysis metrics.

The paper is structured as follows: In section 2 we describe our notion of social events and the annotated data we use for training and testing in our experiments. In section 3, we briefly describe the tree kernel structures used by our system to detect social events in text. Section 4 presents the social network analysis metrics we use to evaluate the quality of the predicted network. In section 5, we present the experiments and results. We discuss some related work in section 6 and conclude in section 7 and mention future directions of research.

## 2 Social Events and Data

In Agarwal et al. (2010), we defined a **social event** as an event in which two people *interact* such that for at least one person, the interaction is **deliberate** or **conscious**. Put differently, at least one person must be aware of the interaction.

[Toujan Faisal], 54, {said} [she] was {informed} of the refusal by an [Interior Ministry committee] overseeing election preparations.

In the above example, the people (or groups of people) involved in social events are *Toujan Faisal* and the *Interior Ministry committee*. There are two social events in this example: one described by the word *said*, in which *Toujan Faisal* is *talking about* the committee, and the other described by the word *informed*, in which *Toujan Faisal* presumably has a mutual interaction with the committee.

We annotated two corpora for social events: 1) The Automatic Content Extraction (ACE) data-set<sup>1</sup> (Agarwal et al., 2010) and 2) the *Alice in Wonderland* data-set<sup>2</sup> (Agarwal et al., 2012).

For each pair of entity mentions in a sentence, if the annotators annotate a social event, we count this as a positive example for the task of social event detection. If no social event is annotated between a pair of entity mentions, we count this as a negative example. Note that we only consider pairs of entity mentions that correspond to different entities; our annotation scheme disregards self-interactions (talking to oneself).

<sup>1</sup>Version: 6.0, Catalog number: LDC2005E18

<sup>2</sup><http://www.gutenberg.org/ebooks/19551>

We use all of ACE data for training and refer to this data-set as **ACE-train**. We use all of *Alice in Wonderland* data for testing and refer to this data-set as **Alice-test**. The distribution of these data-sets is presented in Table 1.

Data-set	# of social events	# of No-event
ACE-train	396	1,101
Alice-test	81	128

Table 1: The distribution of social events in the training and test sets used for experiments

## 3 SINNET: Social Interaction Network Extractor from Text

In Agarwal and Rambow (2010), we presented a preliminary system that extracts social events from news articles. We used Support Vector Machines (SVM) in conjunction with tree kernels for detecting social events between pairs of entities, called target entities, that co-occur in a sentence. Following is a brief description of the tree structures that we used for building our models. We used the Stanford parser’s (Klein and Manning, 2003) phrase structure and dependency tree representations. Of the following tree structures, 1-3 have previously been proposed by Nguyen et al. (2009) for the relation extraction task, while we introduced the fourth structure in Agarwal and Rambow (2010) for social event detection task.

1. PET: This refers to the smallest phrase structure tree that contains the two target entities.
2. Grammatical Relation (GR) tree: This refers to the smallest dependency tree that contains the two target entities. We replace the words (in the nodes of the tree) with their corresponding grammatical roles. For example, in Figure 1, if we replace *Toujan Faisal* by *nsubj*, *54* by *appos*, *she* by *nsubjpass* and so on, we will get a GR tree.
3. Grammatical Relation Word (GRW) tree: We get this tree by adding the grammatical relations as separate nodes between a node and its parent. For example, in Figure 1, if we add *nsubj* as a node between *T1-Individual* and *Toujan Faisal*, add *appos* as a node between *54* and *Toujan Faisal*, and so on, we will get a GRW tree.



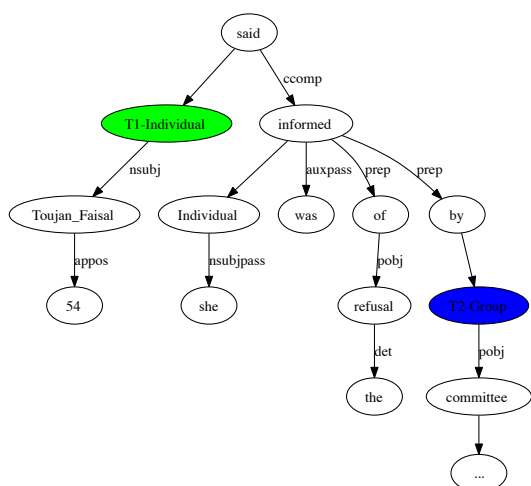


Figure 1: Dependency parse tree for the sentence (in the ACE corpus): *[Toujan Faisal], 54, said [she] was informed of the refusal by an [Interior Ministry committee] overseeing election preparations.*

4. Sequence in GRW tree (SqGRW): This is the sequence of nodes from one target to the other in the GRW tree. For example, in Figure 1, this would be *Toujan\_Faisal nsubj T1-Individual said ccomp informed prep by T2-Group pobj committee*.

We also use combinations of the aforementioned structures. For example, PET\_GR\_SqGRW refers to a kernel that considers a linear combination of three structures (PET, GR and SqGRW) for calculating similarities between examples. We use the Partial Tree kernel, first proposed by Moschitti (2006a), to calculate similarities between these tree structures.

In this paper, we use a Bag of Words Model (BOW) as a baseline. In the BOW model, each sentence is represented as a vector of three feature spaces. The first feature space encodes the presence and absence of words between the start of sentence and the start of the first target entity mention. The second feature space encodes the presence and absence of words between the end of the first target entity mention and the start of the second target entity mention. The third feature space encodes the presence and absence of words between the end of the second target entity mention and the end of the sentence. This feature space has previously been used by GuoDong et

al. (2005) for the relation extraction task on ACE. We use stemming and remove stop words from our feature space.

## 4 Social Network Analysis Metrics

In this section we briefly discuss some of the most popular social network analysis (SNA) metrics used by researchers to mine information from networks. We evaluate the social network extracted by our system with the gold network using these metrics. At a broad level, SNA researchers are interested in measuring importance of nodes in the network and in finding community structures in the network. To measure the importance of nodes, they use the notion of centrality. Following are the centrality measures that are often used in the literature (Freeman, 1979):

1. **Degree centrality** of a node in the network measures the number of incoming and outgoing links from the node. Degree centrality is viewed as an index of a node's *communication activity*.
2. **Betweenness centrality** of a node in the network measures the frequency with which a point falls between pairs of other nodes on the shortest paths connecting them. Nodes with high betweenness centrality are strategically located on the communication paths linking pairs of others, thus having the potential of influencing the group by withholding or distorting information (Bavelas, 1948; Shaw, 1954; Shimbel, 1953).

Another aspect of social networks that SNA researchers are interested in has to do with finding communities in the network and structural properties of networks. Following are some basic metrics used for this task:

1. **Graph density**: The density of a graph is the ratio of the number of edges to the number of possible edges. This measures how close the network is to being complete.
2. **Connected components**: a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by some path. The number of connected components is an indication of the overall connectivity of the network.

3. **Triads:** A triad is a set of three parties connected pair-wise to each other. In his seminal work, Simmel (1950) argued that triads are a fundamental unit of sociological analysis. He argued that three actors in a triad may allow qualitatively different social dynamics that cannot be reduced to individuals or dyads.

## 5 Experiments and Results

We present experiments and results for two tasks: social event detection and social network extraction. We use the same system for both tasks; the first task is an intrinsic evaluation of our system, while the second task presents an extrinsic evaluation of our system. In the following subsections, we describe the individual tasks, their experimental set-up followed by a discussion of results.

### 5.1 Social Event Detection

Task description: Given a pair of entity mentions in a sentence, we evaluate how well we identify the occurrence of a social event between these two entities. This is a binary task with two classes: presence/absence of a social event. We evaluate using F-measure on the presence of social events.

Tree structure	P	R	F
BOW	34.62	77.78	47.91
PET	58.54	59.26	58.90
GRW	49.14	70.37	57.87
SqGRW	49.59	74.07	59.41
PET_GR	56.32	60.49	58.33
PET_GR_SqGRW	56.82	61.73	59.17
GR_SqGRW	54.37	69.14	60.87
GRW_SqGRW	51.30	72.84	60.20
GR_GRW_SqGRW	50.47	66.67	57.45

Table 2: Results for training on ACE-train and testing on Alice-test. P refers to Precision, R refers to Recall, F refers to F1-measure. Terminology used for the tree structures is explained in detail in Section 3.

Experimental set-up: For all our experiments, we use the SVM-Light-TK package (Moschitti, 2006b) for training models. We use the default parameters of the package to avoid over-fitting. Since we are interested in knowing how well we do at finding the social events, we report Precision, Recall and F-measure of the class of interest (the minority class) instead of % accuracy. For

training, we set the  $-j$  parameter of the package to the ratio of the number of negative examples to the number of positive examples in the training data-set. The  $-j$  parameter assigns a weight to the minority class. Since the SVM optimizes for accuracy, if we do not set this parameter at the time of training, the learner may learn a trivial hyperplane classifying all the examples as negative (the majority class), thus achieving a high accuracy. By assigning a weight to the examples in the minority class, we increase the cost of mis-classifying these examples, thus forcing the learner to find a non-trivial hyperplane. We use the model trained just on bag-of-words (BOW) as a baseline.

Discussion of results: Table 2 presents the results for models trained on ACE-train and tested on Alice-test. We use the tree kernel structure combinations described in section 3. The results show that building a model using tree kernels outperforms the bag-of-words baseline model by an absolute 10% F1-measure. This difference is statistically significant with  $p < 0.05$ .

### 5.2 Social Network Extraction

In this section, we provide results to establish that the un-weighted network extracted using social event detection models is *close* to the true underlying network.

Task description and experimental set-up: Using our social event detection models, we build an un-weighted network of entities in *Alice in Wonderland*. For this task, we report the *distance* between the SNA metrics in the predicted and gold networks. Table 3 summarizes the metrics we use for our evaluation and elaborates on the meaning of *distance*. We use two baseline systems for our evaluation:

1. **B-Simple:** For this baseline, we create a network by linking all pairs of entity mentions (of different entities) that appear in the same sentence.
2. **B-BOW:** This is the network extracted by building a model that uses bag-of-words features for training.

Discussion of results: Table 4 shows results for the SNA metrics (Section 4) for the kernel combinations used to extract a network from *Alice in Wonderland*. The terminology used in Table 4 is explained in Table 3.

Symbol	Name and explanation
P, R, F, %A	Precision, Recall, F1-measure and Accuracy
p	McNemar’s two-sided p-value significance test. We linearize the adjacency matrix of the predicted and gold network and test if these two vectors are significantly different.
D, B	Degree and Betweenness centrality. For each of these centrality measures, we find the centrality of nodes in the network, represented by a vector $\vec{v}$ , where the $i^{th}$ component of the vector is the centrality of the $i^{th}$ node. We then calculate the Euclidean distance between the predicted and gold vectors, which measures the difference in degree centralities of the nodes in the two networks.
S, CC, T	Network density, number of connected components and number of triads respectively. For these measures, we calculate the difference between the gold and the predicted network. For example, a value of 6 in the column labeled CC and the row labeled Alice in Table 2 is the difference in number of connected components found in the predicted network and the gold network.

Table 3: Terminology used to present results in Table 4

System	Stats					Centrality		Community		
	P $\uparrow$	R $\uparrow$	F $\uparrow$	%A $\uparrow$	p $\uparrow$	D $\downarrow$	B $\downarrow$	S $\downarrow$	CC $\downarrow$	T $\downarrow$
B-Simple	0.40	1.00	0.57	97.58	0.0000	19.67	0.57	0.0245	23	439
B-BOW	0.47	0.87	0.61	98.15	0.0000	13.00	0.34	0.0145	14	165
PET	<b>0.77</b>	0.65	<b>0.71</b>	<b>99.12</b>	0.12	<b>6.93</b>	0.15	0.0026	2	<b>0</b>
GRW	0.61	0.68	0.65	98.78	0.38	8.31	0.10	0.0019	1	49
SqGRW	0.64	<b>0.81</b>	<b>0.71</b>	98.93	0.01	8.77	0.11	0.0045	6	34
PET_GR	0.72	0.60	0.66	98.96	0.15	7.75	0.12	0.0026	2	28
PET_GR _SqGRW	0.72	0.65	0.68	99.01	0.41	7.87	0.10	0.0016	3	18
GR_SqGRW	0.67	0.70	0.68	98.93	<b>0.75</b>	8.77	<b>0.06</b>	<b>0.0008</b>	1	23
GR_GRW _SqGRW	0.63	0.68	0.66	98.83	0.55	8.31	0.10	0.0013	<b>0</b>	6

Table 4: Results comparing the two baseline systems (B-Simple and B-BOW) with the models trained on the tree kernel structures discussed in Section 3.  $\uparrow$  means greater value is better.  $\downarrow$  means lesser value is better. Network density of the gold network is 0.0166. The number of connected components in the gold network are 34. The number of triads in the gold network are 103.

Table 4 shows that all the systems trained using tree kernels are better than the two baselines across all SNA metrics. In terms of F-measure, both the baselines perform significantly worse than the tree kernel based systems. In terms of p-value, both the baselines are significantly different from the gold network, whereas none of the tree kernel based systems are significantly different from the gold network. In terms of the distance between the vectors of centrality measures (D, B) for the predicted and gold network – the distance is larger for the baselines, which means that the difference in centrality measures of nodes in the baseline system and the gold network is larger than the differ-

ence in centrality measures of nodes in the other systems and the gold network. The difference in network densities of the baseline and gold network is also larger than the difference in network densities (S) of the other systems and the gold network. The same is the case with the number of connected components (CC) and the number of triads (T). Using these results, we conclude that the network predicted by our system that uses tree kernels performs well in terms of extracting an unweighted, undirected network from *Alice in Wonderland*. In particular, the tree structure derived from the phrase structure tree (PET) performs the best on most of the SNA metrics.

## 6 Literature Survey

With the advent of the internet and social media, researchers have got access to different forms of communication such as Email (Klimt and Yang, 2004; Rowe et al., 2007), online discussion threads (Hassan et al., 2012), Slashdot, Epinions, and Wikipedia (Jure Leskovec and Kleinberg, 2010). There have also been approaches of extracting networks based on Information Retrieval techniques – Jing et al. (2007) extract a network from conversational speech data. The events they are interested in are custody, death, hiding, liberation, marriage, migration, survival and violence. Tang et al. (2008) aim at extracting and mining academic social networks. Aron Culotta and McCallum (2004) extract social networks and contact information from email and the Web. Mori et al. (2006) mine networks based on the collective context in which entities appear.

Our notion of social network is different from the aforementioned work. We are interested in extracting *interaction* networks from unstructured text. In terms of our goals, our work is closest to the work by Elson et al. (2010) and He et al. (2013). Elson et al. (2010) and He et al. (2013) are also interested in extracting a social network from literary texts. However, they restrict their definition of *interaction* to interactions that are signaled by quotation marks. Our system does not have this limitation and is therefore able to extract interaction links appearing even in reported speech (non-dialogue text).

## 7 Conclusion and Future Work

In this paper, we have addressed the problem of extracting a social network from literary narrative text. We have used our previous system that detects social events to extract a network from *Alice in Wonderland*. This system was trained on news articles and has never been tested out of domain. Our evaluation on *Alice in Wonderland* has two components: a standard intrinsic evaluation in terms of the detection of social events, and an extrinsic evaluation which measures how well the un-weighted network formed by the extracted social events mirrors the gold social network. For the extrinsic evaluation, we use various network measures such as centrality or density. We show that while we achieve an F-measure of about 61% on the intrinsic evaluation, our extracted network is not statistically distinguishable from the gold net-

work according to the various network measures.

In the future, we will apply our system to more literary texts. We are currently acquiring annotations on 19th century novels such as *Emma* by Jane Austen. We will also apply our system to other genres such as historical documents.

## Acknowledgments

This paper is based upon work supported in part by the DARPA DEFT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

## References

- Apoorv Agarwal and Owen Rambow. 2010. Automatic detection and classification of social events. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1034, Cambridge, MA, October. Association for Computational Linguistics.
- Apoorv Agarwal, Owen C. Rambow, and Rebecca J. Passonneau. 2010. Annotation scheme for social network extraction from text. In *Proceedings of the Fourth Linguistic Annotation Workshop*.
- Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. 2012. Social network analysis of *alice in wonderland*. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 88–96, Montréal, Canada, June. Association for Computational Linguistics.
- Ron Bekkerman Aron Culotta and Andrew McCallum. 2004. Extracting social networks and contact information from email and the web. In *First Conference on Email and Anti-Spam (CEAS)*.
- A. Bavelas. 1948. A mathematical model for group structures. *Human Organization*, 7:16–30.
- Stephen P. Borgatti and Rob Cross. 2003. A relational view of information seeking and learning in social networks. *Management science*.
- U. Brandes, J. Raab, and D. Wagner. 2001. Exploratory network visualization: Simultaneous display of actor status and connections. *Journal of Social Structure*.
- Rob Cross, Andrew Parker, and Laurence Prusak. 2001. Knowing what we know:-supporting knowledge creation and sharing in social networks. *Organizational Dynamics*.
- David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147.

- Linton C. Freeman. 1979. Centrality in social networks conceptual clarification. *Social Networks*, 1 (3):215–239.
- Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics*.
- P. Hage and F. Harary. 1983. *Structural models in anthropology*. Cambridge University Press.
- Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012. Extracting signed social networks from text. In *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, TextGraphs-7 '12, pages 6–14, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. *The 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.
- Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2007. Extracting social networks and biographical facts from conversational speech transcripts. *45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic*.
- J. C. Johnson. 1994. Anthropological contributions to the study of social networks: A review. *Advances in social network analysis: Research in the social and behavioral sciences*.
- Daniel Huttenlocher Jure Leskovec and Jon Kleinberg. 2010. Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.
- Bryan Klimt and Yiming Yang. 2004. Introducing the enron corpus. In *proceedings of the First Conference on Email and Anti-Spam (CEAS)*.
- D. Knoke. 1990. *Political Networks: The structural perspective*. Cambridge University Press.
- Laura M. Koehly and Victoria A. Shivy. 1998. Social network analysis: A new methodology for counseling research. *Journal of Counseling Psychology*.
- Franco Moretti. 2005. *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso.
- Junichiro Mori, Takumi Tsujishita, Yutaka Matsuo, and Mitsuru Ishizuka. 2006. Extracting relations in social networks from the web using similarity between collective contexts. In *The Semantic Web-ISWC 2006*, pages 487–500. Springer.
- Alessandro Moschitti. 2006a. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*.
- Alessandro Moschitti. 2006b. Making tree kernels practical for natural language learning. In *Proceedings of European chapter of Association for Computational Linguistics*.
- Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. *Conference on Empirical Methods in Natural Language Processing*.
- Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. 2007. Automated social hierarchy detection through email network analysis. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117.
- R. Sanjek. 1974. What is social network analysis, and what it is good for? *Reviews in Anthropology*.
- S. B. Seidman. 1985. Structural consequences of individual position in nondyadic social networks. *Journal of Mathematical Psychology*.
- Marvin E. Shaw. 1954. Group structure and the behavior of individuals in small groups. *The Journal of Psychology*, 38(1):139–149.
- Alfonso Shimbel. 1953. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15(4):501–507.
- Georg Simmel. 1950. *The Sociology of Georg Simmel*. The Free Press, New York.
- Malcolm K. Sparrow. 1991. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social networks*.
- Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM.
- Noel M. Tichy, Michael L. Tushman, and Charles Fombrun. 1979. Social network analysis for organizations. *Academy of Management Review*, pages 507–519.

# Using the Web to Train a Mobile Device Oriented Japanese Input Method Editor

Xianchao Wu, Rixin Xiao, Xiaoxin Chen

Baidu Inc.

wuxianchao@gmail.com, {xiaorixin, chenxiaoxin}@baidu.com

## Abstract

This paper describes the construction of a Japanese Input Method Editor (IME) system for mobile devices, using the large-scale Web pages. We provide the training process of our IME model, n-pos model for local Kana-Kanji conversion and n-gram model for online cloud service. Especially, we propose an online algorithm of mining new compound words, together with the detailed post-filtering process to prune the billion level entries to be used in mobile services. Experiments show that our IME system outperforms two state-of-the-art Japanese IME baselines. We have released our system in a completely free form<sup>1</sup> and the system is currently used by millions of users.

## 1 Introduction

Mobile devices such as smart phones and tablet PCs are used by billions of users. For example, Google's Android system has obtained more than 900 million<sup>2</sup> active devices till May 2013. In this paper, we use large-scale Web pages to train a Japanese IME system for mobile devices.

Languages such as Chinese and Japanese can not be typed directly using Latin keyboards. This is because there are only 26 English letters in a Latin keyboard, yet the number of Chinese characters are in tens thousands level. Furthermore, by connecting several Chinese characters together, the number of yielded words and frequently used phrases/idioms are in million level. Thus, language-dependent Input Method Editor (IME) systems are indispensable which maps from combinations of English letters to Chinese char-

acters/words/phrases and Japanese Kanji/Kana sequences.

For example, suppose we want to input a Japanese verb “炒める” (means “fry”, “炒” is a Japanese Kanji character, “める” are two Japanese Kana characters). We first need to know the Kana pronunciation of the Kanji character “炒”, which is “いた”. That is, the verb is pronounced as “いためる”. Then, we need to know the mapping from English letters to Japanese Kanas. Here, “い”, “た”, “め”, “る” respectively correspond to “i”, “ta”, “me”, “ru”, which can be directly typed by using the Latin keyboards. This mapping (e.g., from “i” to “い”) is unique and predefined already. Thus, the real challenge for constructing an IME system for Japanese is to provide the most reasonable Kanji sequence from a given Kana sequence:

- one Kanji character can have several correct Kana pronunciations (e.g., “炒” can be pronounced as “いた”, “しょう”, “そう”, etc.);
- one Kana sequence corresponds to numerous Kanji candidates (such as “いためる” for “炒める”, “痛める” (pain), etc.);

It is the context that determines the selection of the most reasonable Kanji sequence. For example, “心をいためる” (“心” = heart, “を” is a Japanese particle right follows an argument and before the argument's predicate) requires the Kanji to be “痛める” (heart pain) and “野菜をいためる” needs the Kanji to be “炒める” (fry vegetables). However, it is not a trivial work for modelling the context. That is, how to choice the context from large-scale Web pages such that the context is optimized to be used in a mobile device oriented Japanese IME?

To answer this question, we need to consider the following constraints:

- mobile devices need more strict controlling of CPU and memory usages than laptops;

<sup>1</sup><http://simeji.me/>

<sup>2</sup><http://www.android.com/>

- free wireless services are not supposed to be available anywhere, any time.

Consequently, we have to limit the number of Kana-Kanji entries to be loaded into memory and ensure a high precision of Kana-Kanji conversion even without on-line services (such as cloud input).

## 2 The Model

Our Japanese IME system is constructed based on the n-pos<sup>3</sup> model (Mori et al., 1999; Komachi et al., 2008; Kudo et al., 2011). For statistical Kana-Kanji conversion, we predicate the optimal mixed Kana-Kanji sequence  $\hat{y}$  ( $= w_1 \dots w_n$ ) from the input Hirakana sequence  $\mathbf{x}$ :

$$\hat{y} = \operatorname{argmax}_{\mathbf{y}} P(\mathbf{y})P(\mathbf{x}|\mathbf{y}) \quad (1)$$

$$P(\mathbf{y}) = \prod_{i=1}^n P(w_i|c_i)P(c_i|c_{i-1}) \quad (2)$$

$$P(\mathbf{x}|\mathbf{y}) = \prod_{i=1}^n P(r_i|w_i) \quad (3)$$

As shown in Figure 1, for training this model, we used 2.5TB Japanese Web pages as the training data. We run Mecab<sup>4</sup> with IPA dictionary<sup>5</sup> on Hadoop<sup>6</sup>, an open source software that implemented the Map-Reduce framework (Dean and Ghemawat, 2004), for parallel word segmenting, Part-of-Speech (POS) tagging, and Kana pronunciation annotating. Then, based on maximum likelihood estimation, we estimate:

- $P(c_i|c_{i-1})$ , bi-gram POS tag model;
- $P(w_i|c_i)$ , POS-to-word emission model, from  $c_i$  to a word  $w_i$ ; and,
- $P(r_i|w_i)$ , pronunciation model, from  $w_i$  to its Kana pronunciation  $r_i$ .

There are several lexicons/models to be used in the final IME system. The first is called the basic lexicon. An entry in this lexicon is alike  $\langle w_i^{i+m}, c_i^{i+m}, r_i^{i+m} \rangle$ . Here,  $w_i^{i+m}$  stands for  $m + 1$  words (of  $w_i \dots w_{i+m}$ ). One word  $w_i$  exactly corresponds to one POS tag  $c_i$  and one Kana

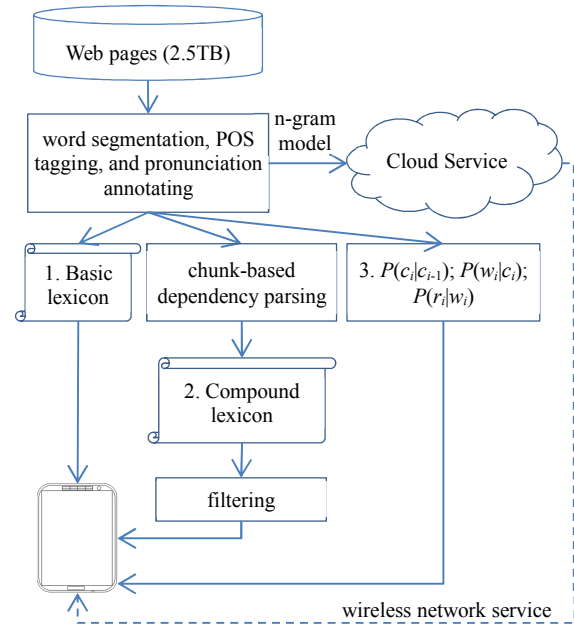


Figure 1: The main process of using the Web to train the IME system.

sequence  $r_i$  as its pronunciation. One word sequence with multiple reasonable POS sequences and/or Kana pronunciations will be stored separately as different entries. This lexicon contains: (1) Japanese words (such as particles, adjectives, adverbs, verbs, nouns, etc.) with the highest frequencies, and (2) the most frequently used idioms which are collected manually by our Japanese language experts.

The second is the compound lexicon which contains new words, collocations, and predicate-argument phrases. As drawn in Figure 1, dependency parsing is performed before mining. Web sentences were parsed by a state-of-the-art chunk-based Japanese dependency parser, Cabocha<sup>7</sup> (Kudo and Matsumoto, 2002a). The mining and filtering process will be introduced in the next section. The motivation of constructing this lexicon is to extract the most important context information, such as the strong constraints among predicates and their arguments. For example, as former mentioned, the pre-predicate arguments such as “心” (heart) or “野菜” (vegetables) with given Kana sequence “をいためる” will determine which predicate verb to choose, “痛める” or “炒める”.

The third is the n-pos model with three kinds of probabilities which are used during decoding, i.e., searching the n-best  $\mathbf{y}$ s from a given input Kana

<sup>3</sup>n-pos model is short for n-gram part-of-speech model

<sup>4</sup><https://code.google.com/p/mecab/>

<sup>5</sup><http://code.google.com/p/mecab/downloads/detail?name=mecab-ipadic-2.7.0-20070801.tar.gz>

<sup>6</sup><http://hadoop.apache.org/>

<sup>7</sup><http://code.google.com/p/cabocha/>

sequence  $\mathbf{x}$ .

Finally, we train a 4-gram language model on surface word level and construct a cloud Kana-Kanji conversion service through wireless network communication between a mobile device and the cloud. The only difference with former n-pos model is the factorization of  $P(\mathbf{y})$ :

$$P(\mathbf{y}) = \prod_{i=1}^n P(w_i | w_{i-1}, w_{i-2}, w_{i-3}) \quad (4)$$

The first three lexicons/models are stored in the mobile devices to be accessed during Kana-Kanji decoding using Equation 1. The final 4-gram language model is estimated in a different way from the n-pos model. Thus, we are forced to interpolate cloud's m-best Kanji candidates into local mobile device's n-best Kanji candidates. We perform duplicated candidate removing before interpolating. Possible methods includes:

- insert the cloud candidates into fixed positions, e.g., from the second position to the m+1 position of the local n-best list; or,
- upload the local n-best candidates to the cloud and then use the 4-gram language model to compute the candidates' language model scores; or,
- download POS tags of the m-best candidates from the cloud and locally compute their scores under the local n-pos model.

The first method is the simplest without a large usage of the wireless network. The later two methods make the direction comparison of cloud and local candidates yet possibly take a large usage of the network. For simplicity, we choose the first method (e.g., taking cloud result as the first candidate) in our IME system.

### 3 Compound Word Mining and Filtering

#### 3.1 Mining Process

The basic lexicon used in our Japanese IME system is short at capturing new words and phrases, which are appearing everyday in the latest Web pages. For example, person names, technical terms and organization names are newly created and used in Web pages such as news, blogs, question-answering systems. We argue it is essential for the IME system to regularly update

its compound lexicon to cover these new and hot words/phrases.

Alike the format of the basic lexicon, entries with  $m + 1$  ( $m$  differs among the entries) words in the compound lexicon is also triples of  $\langle w_i^{i+m}, c_i^{i+m}, r_i^{i+m} \rangle$ . In this paper, we mine three types of new compound words, together with their pronunciations from Japanese Web pages:

- words, which are combinations of single characters and shorter words (e.g., “副/ふく 垢/あか” = “secondary (twitter) account”);
- collocations, which are combinations of words (e.g., “ドバイ ショック” = “Dubai (debt) crisis”). Here, Japanese collocations are allowed to include Kanjis, Katakanas and Hirakanas. Different from many former researches (Manning and Schütze, 1999; Liu et al., 2009) which only mine collocations of two words, we do not limit the number of words in our “collocation” lexicon; and,
- predicate-argument phrases, which are combinations of chunks constrained by semantic dependency relations (e.g., “心 を 痛める” = heart pain).

New words and collocations are mined from single chunks in the dependency trees generated by Cabocha. This mining idea is based on the fact that an Japanese morphological analyser (e.g., Mecab) tends to split one out-of-vocabulary (OOV) word into a sequence of known Kanji characters, and most of these known Kanji characters are annotated to be notional words. Consequently, Cabocha, which takes words/characters and their POS tags as features for discriminative training using a SVM model (Kudo and Matsumoto, 2002b), can still *correctly* tend to include these single-Kanji-character words into one chunk. Thus, we can re-combine the wrongly separated pieces into one (compound) word.

Predicate-argument phrases are mined from adjacent chunks with dependency relations. Since Japanese is a Subject-Object-Verb (SOV) language, the predicate frequently follows its subject/object arguments.

Figure 2 and 3 give the distributions (number of words per compound word vs. the frequency of compound words in a similar number of words) of single/double chunk lexicons (without any filtering yet). For new words and collocations mined



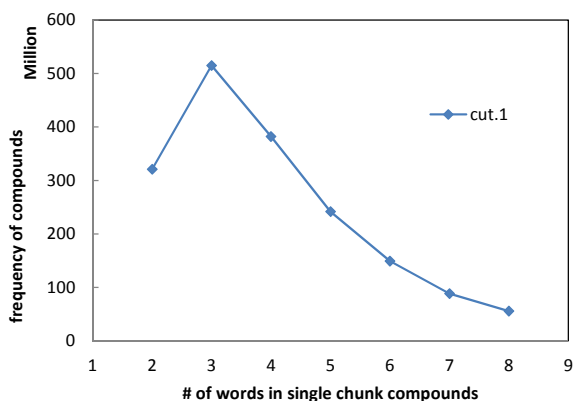


Figure 2: The distributions of the number of words per compound in single chunk lexicon, mined from the 2.5TB web data.

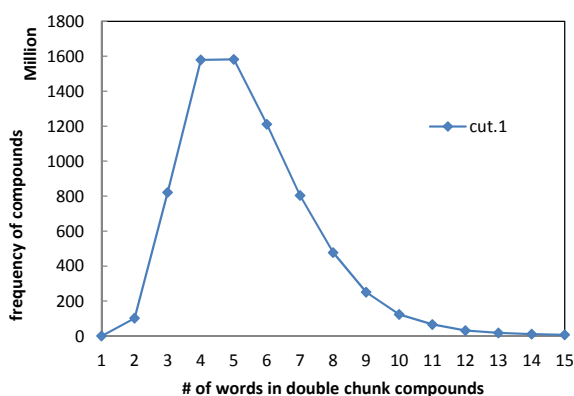


Figure 3: The distributions of the number of words per predicate-argument phrases, mined from the 2.5TB web data.

from single chunks, we limit the number of words frequently ranges from 2 to 8. For predicate-argument phrases, most number (of words) is in the interval of [2, 11].

Frequency information is used for pruning the compound entries to be finally used in mobile devices. The mining algorithm can be performed in an online way. For one aspect, we can timely crawl the latest Web pages and execute the mining process. The frequencies of lastly mined entries can be simply accumulated to the existing entries. On the other hand, we allow the users to upload their input logs to the cloud and execute the mining process to extract single user oriented personally entries. Again, the frequencies of similar words/phrases are simply accumulated.

Recall the n-pos model and the n-gram language model. Since all the probabilities are estimated in a maximum likelihood way, we can simply update the probabilities in these models by accumu-

lating frequencies to similar words/phrases. Thus, we say that our IME system is self-growing as the Web becoming larger and users using it longer.

### 3.2 Filtering Process

After successful mining of the compound lexicon, it is still challenging to prune it to be used in mobile devices with limited computing ability and memory. The trade-off is that, we have to maintain a good enough local lexicon yet with extremely limited number of entries. We use the following algorithms step by step for filtering:

- use the likelihood ratio method as described in (Manning and Schütze, 1999);
- use the LH score as described in (Okazaki and Ananiadou, 2006);
- use the log file of the cloud service; and,
- use hand-made deep filtering rules.

We hereafter describe these filtering strategies. Likelihood ratio is an approach to hypothesis testing, which has been proved to be appropriate for sparse data (Manning and Schütze, 1999). That is, even for candidate phrases will relatively low frequencies, if they share a strong relation with each other, then they are still possibly kept. Likelihood ratio is simply a number that tells us how much more likely one hypothesis is than the other one:

- Hypothesis 1.  $P(w_2|w_1) = p = P(w_2 | -w_1)$ ;
- Hypothesis 2.  $P(w_2|w_1) = p \neq P(w_2 | -w_1)$ .

The first hypothesis judges if the occurrence of word  $w_2$  is *independence* with word  $w_1$ , and the second hypothesis is about *dependence* which is good evidence for an interesting collocation candidate. The computing of  $\log\lambda = \log(L(H_1)/L(H_2))$  exactly follows the definitions in (Manning and Schütze, 1999). Deal to short of space, we skip the detail here. Note that for phrases with more than two words (e.g., three words), we separately take the first/last two words as one unit, and the other word as another unit. Then, if and only if both  $w_1w_2, w_3$  and  $w_1, w_2w_3$  are collocation candidates, then  $w_1w_2w_3$  is taken as one reasonable collocation.

After likelihood ratio based filtering, we checked the remaining entities and found too

many nested entries. For example, even both  $w_1w_2w_3$  and  $w_1w_2$  were kept in the final compound lexicon, only one of them was judged manually to be the correct one. Dealing with this problem, we use the LH score formula as described in (Okazaki and Ananiadou, 2006). we reuse the heuristic LH formula to compute the collocation likelihood  $LH(c)$  for a candidate  $c$ :

$$LH(c) = \text{freq}(c) - \sum_{t \in T_c} \text{freq}(t) \times \frac{\text{freq}(t)}{\sum_{t \in T_c} \text{freq}(t)}.$$

Here,  $c$  is a Kanji (sub-)sequence candidate;  $\text{freq}(c)$  denotes the frequency of co-occurrence of  $c$  with the final/first word(s) of the phrases; and  $T_c$  is a set of nested Kanji sequence candidates, each of which consists of a preceding (or, succeeding) Kanji or Kana character followed by (or, follows) the candidate  $c$ . This LH score can be computed in left-to-right (i.e., taking the former one or more words as no-changing words and seek the word list that follows these former words) direction or right-to-left direction. For example, for left-to-right computing, we can collect all the phrases starts with the similar word  $w_1$  and then collect all the compound entries start with  $w_1$ . Then, after computing LH score, we can limit the number of entries start with  $w_1$ .

Even after these two automatic filtering algorithms, we still find there are too many entries remaining in the compound lexicon. The third step of filtering is the usage of the cloud log file which stores the entries that users uploads to the cloud. This filtering strategy is to only keep those entries whose Kana pronunciations were found in the log file. The consideration is to connect the Web to the real requirements of the users.

Finally, we manually check the remaining lexicon and construct deep filtering rules. For example, entries that starts with “ない”, “等” are pruned out; entries with POS tags of “particles”, “auxiliary verbs” are pruned out. Note that, this manual checking is performed before the final lexicon is generated. Filtering rules are constructed after this manual checking step and further used for filtering the test set as can be find in the next section (22 entries were filtered from the 5K entries in the test set).

## 4 Experimental Results

We have described in detail the training and filtering process for constructing lexicons and models.

Systems	top-1	top-6	top-12	Missing Words
Baseline1	84.91	89.11	89.31	532
Baseline2	82.64	94.23	94.80	112
IME-basic	81.36	85.82	85.82	705
+ compound	85.78	91.22	91.30	431
+ cloud (1st)	88.99	94.98	96.44	41

Table 1: The top-1/6/12 precisions (%) of the baselines and our IME system under several configurations. Here, IME-basic stands for our IME system with only the basic lexicon; + compound stands for the system together with the basic lexicon and the compound lexicon; + cloud stands for taking cloud’s best candidate as IME’s first candidate.

In terms of the decoding algorithm, we use beam searching for n-best Viterbi decoding (Huang and Chiang, 2005). The training data is a 2.5TB Japanese Web page set. Our basic lexicon contains around 100k entries, while the compound lexicon is limited to contain around 50k entries. No limitation is set to the 4-gram language model running in the cloud.

Our test set contains 4,978 Kana-Kanji entries of frequently used word, idioms, and phrases. The entries of this test set comes from the following three lexicons/corpora:

- (partial) “JDMWE” (Japanese Dictionary of Multi-Word Expressions) (Shudo et al., 2011) lexicon with 2,169 entries;
- “Nagoya” compound word lexicon<sup>8</sup> with 3,628 entries such as idioms;
- 16,611 long form words in the “BCCWJ” (Balanced Corpus of Contemporary Written Japanese) corpus (Maekawa, 2008).

We then retrieve each entry in these three lexicons using Google<sup>9</sup> and only keep the top 5K entries with higher frequencies. After obtaining the 5K entries, we perform manually constructed deep filtering rules (which have been used during training) and remove 22 entries which are judged to be not suitable to be taken as collocations with complete meaning.

We use top-n precisions  $P_n$  to evaluate the accuracy of the IME systems. We use  $\langle k_m, r_m \rangle$  to

<sup>8</sup><http://kotoba.nuee.nagoya-u.ac.jp/jc2/base/list>

<sup>9</sup><https://www.google.co.jp/>

express one entry in the test set, where  $m$  ranges from 1 to  $M$ ,  $k_m$  is the Kana input and  $r_m$  is the Kanji reference.

$$P_n = \frac{\sum_{m=1}^M \{\delta(r_m, \text{IME}_n(k_m))\}}{M} \quad (5)$$

Given one  $k_m$ ,  $\text{IME}_n(k_m)$  generates the  $n$ -best Kanji candidate for  $k_m$ . The  $\delta()$  function is defined as follows:

$$\delta(r_m, \text{IME}_n(k_m)) = \begin{cases} 1 & \text{if } r_m \in \text{IME}_n(k_m), \\ 0 & \text{otherwise.} \end{cases}$$

When  $n$  takes 1,  $P_1$  is equivalent to the traditional definition of precision.

Table 1 shows the top- $n$  precisions and the number of missing words of two state-of-the-art Japanese IME baseline systems and our IME system under several configurations of lexicons/models. Both the baseline systems and our IME systems are in mobile device versions.

Here, baseline1<sup>10</sup> is a commercial Japanese IME system whose lexicon contains around 200k entries. This baseline system is constructed by using statistical methods on relatively a small-scale training data and a lot of hand-made Kana-Kanji conversion rules. Deal to resource limitation, we could not obtain further detailed technical information of this system and can only buy one copy and test it in an open testing way.

The second baseline IME system (Kudo et al., 2011), baseline2<sup>11</sup>, is constructed in a statistical way by using the large-scale Japanese Web pages as the training data. N-pos model is also the major model supporting its training and decoding algorithms. This system can be freely obtained.

From the table, we have the following observations:

- when only using the basic lexicon, our IME system is worse than both of the baselines;
- when the compound lexicon is appended, the top-1 precision of our IME system is better than baselines, yet top-6/12 precisions are still not good (by checking the lexicon size of baseline2, we found that around 300k to 400k entries were contained. Yet there are only around 100k+50k entries in our basic/compound lexicons);

<sup>10</sup>[http://www.justsystems.com/jp/products/atok\\_android/](http://www.justsystems.com/jp/products/atok_android/)

<sup>11</sup><https://play.google.com/store/apps/details?id=com.google.android.inputmethod.japanese>

Systems	top-1	top-6	top-12	Missing Words
IME	76.12	82.05	82.05	224
+ log	79.41	87.74	87.82	152
improves	3.29	5.69	5.77	-72

Table 2: The top-1/6/12 precisions (%) of our IME system under several configurations. Here, IME stands for the system using basic and compound lexicons; + log stands for appending compound entries mined from users' log.

- finally, by using cloud service, the top- $n$  precisions are significantly better than two baselines.

We did another experiment for testifying the “online” ability of our IME system. The training data is the users' logs. We used these logs (of during two months) to extract compound words and append them to existing compound lexicons. There are 6k entries appended. The testing data (which contains 1,248 entries) is a set of compound words using logs of the latest three days.

Table 2 shows the changes of top-1/6/12 precisions by appending the compound entries mined from users' log. We observe that the precisions are improved 3.29% to 5.77%. These improvements show evidence that the IME system can grow itself in an online way with more data and more users.

## 5 Conclusion

We have described the construction of a Japanese Input Method Editor (IME) system for mobile devices, using 2.5TB Web pages. We provided the training process of our IME model, n-pos model for local Kana-Kanji conversion and n-gram model for online cloud service. In particular, we described an online algorithm of mining new compound words, together with the detailed post-filtering process to prune the billion level entries to be used in mobile services. Experiments showed that our IME system outperforms two state-of-the-art Japanese IME baselines. We have released our system in a completely free form and the system has been downloaded by more than 5 million users and is currently used by users in million level<sup>12</sup>.

<sup>12</sup><https://play.google.com/store/apps/details?id=com.adamrocker.android.input.simeji>

## References

- Jeffrey Dean and Sanjay Ghemawat. 2004. Mapreduce: simplified data processing on large clusters. In *Proceedings of OSDI*.
- Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of IWPT*.
- Mamoru Komachi, Shinsuke Mori, and Hiroyuki Tokunaga. 2008. Japanese, the ambiguous, and input methods (in japanese). In *Proceedings of the Summer Programming Symposium of Information Processing Society of Japan*.
- Taku Kudo and Yuji Matsumoto. 2002a. Japanese dependency analysis using cascaded chunking. In *Proceedings of Co-NLL*, pages 63–69.
- Taku Kudo and Yuji Matsumoto. 2002b. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL-2002*, pages 63–69. Taipei, Taiwan.
- Taku Kudo, Taiyaki Komatsu, Toshiyuki Hanaoka, Jun Mukai, and Yusuke Tabata. 2011. Mozc: A statistical kana-kanji conversion system (in japanese). In *Proceedings of Japan Natural Language Processing*, pages 948–951.
- Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. 2009. Collocation extraction using monolingual word alignment method. In *Proceedings of EMNLP*, pages 487–495, Singapore, August.
- Kikuo Maekawa. 2008. Compilation of the kotonoha-bccwj corpus (in japanese). *Nihongo no kenkyu (Studies in Japanese)*, 4:82–95.
- Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, May.
- Shinsuke Mori, Masatoshi Tsuchiya, Osamu Yamaji, and Makoto Nagao. 1999. Kana-kanji conversion by a stochastic model (in japanese). *Journal of Information Processing Society of Japan*, 40(7).
- Naoaki Okazaki and Sophia Ananiadou. 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(22):3089–3095.
- Kosho Shudo, Akira Kurahone, and Toshifumi Tanabe. 2011. A comprehensive dictionary of multiword expressions. In *Proceedings of ACL-HLT*, pages 161–170, Portland, Oregon, USA, June.

# A Novel Approach Towards Incorporating Context Processing Capabilities in NLIDB System

Arjun R. Akula, Rajeev Sangal, Radhika Mamidi

Language Technologies Research Center,  
IIT Hyderabad, India.

arjunreddy.akula@research.iiit.ac.in,  
{sangal, radhika.mamidi}@iiit.ac.in

## Abstract

This paper presents a novel approach to categorize, model and identify contextual information in natural language interface to database (NLIDB) systems. The interactions between user and system are categorized and modeled based on the way in which the contextual information is utilized in the interactions. A relationship schema among the responses (user and system responses) is proposed. We present a novel method to identify contextual information in one specific type of user-system interaction. We report on results of experiments with the university related queries.

## 1 Introduction

Natural Language Interface to Database (NLIDB) systems allow the users to query databases in a natural language (Androustopoulos et al., 1995; Meng and Wang, 2001; Popescu et al., 2003; Stratica et al., 2005; Li et al., 2005; Giordani, 2008; Giordani and Moschitti, 2009; Gupta et al., 2012). Although NLIDB systems are able to answer a wide range of natural language queries (NL queries), they are not used much in commercial applications. One of the main reasons for the less acceptance of these systems in real-time applications is that they lack robust context processing capabilities (Bertomeu et al., 2006). Currently there is very little work which explicitly aims to investigate the role of context processing capabilities in NLIDB systems. However, the importance of context processing capabilities has been explored extensively in Question Answering systems (Chai and Jin, 2004; Kato et al., 2004; Kirschner and Bernardi, 2007; Negri and Kouylekov, 2007; Kirschner and Bernardi, 2010).

Users often fail to express their intention (information need) in a single NL query (user re-

sponse) (Bertomeu et al., 2006). Hence to answer a sequence of related NL queries, NLIDB systems should keep track of contextual information. NLIDB systems which do not use contextual information (non-contextual NLIDB) fail to completely capture the user's intention.

**U1:** How many third year students have registered for Robotics course in Monsoon 2011?

**S1:** 12

**U2:** How many got 'A' grade?

**S2:** 2

**U3:** Show the name of the professor who guides the student 'Newton'

**S3:** Prof. Einstein

**U4:** Did he teach any course?

**S4:** No

Figure 1: An example of context based user-system interaction

For example, let us consider a user-system interaction shown in Figure 1. User responses are represented as U1, U2, etc. and system responses are represented as S1, S2, etc. In this example, to interpret U2, information present in the preceding query U1 is needed. That means information present in U1 is the contextual information for U2. Query U3 does not depend on the information present in preceding queries. Semester name 'Monsoon 2011' present in U1 and the professor name 'Einstein' present in S3 are needed to interpret U4.

### 1.1 Background

In a semantic template based non-contextual NLIDB system (Gupta et al., 2012), the main stages involved in extracting answers (system's response) from the database are shown in Figure 2. At the syntactic analysis stage, the linguistic information is extracted from the NL query. At the semantic analysis stage, entities, attributes and the

values to these attributes are identified by using the output of the syntactic analysis module and semantic templates. At the query processing stage, entities identified in the semantic stage are mapped onto the domain conceptual model based on an entity relationship graph (ER graph) and a shortest path in the ER graph connecting them is computed. SQL (Structured Query Language) query is generated using the path obtained and the SQL query is later executed to produce results.

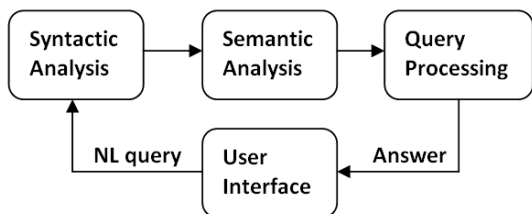


Figure 2: NLIDB system without context processing capabilities

In our approach, the context processing capabilities are incorporated into a non-contextual NLIDB system without disturbing the internal functioning of the existing modules of the system as shown in Figure 3.

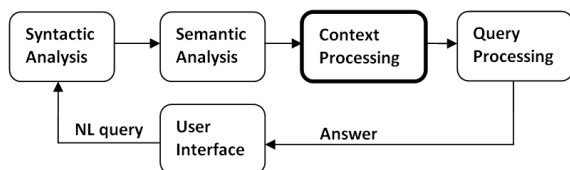


Figure 3: Context based NLIDB system

## 2 Related Work

Chai and Jin (2004) and Sun and Chai (2007) investigate the role of discourse modeling to track contextual information in interactive Question Answering systems. They analyzed the relations between user’s responses and proposed models based on centering theory to identify the contextual information. However, the above mentioned models fail to utilize system’s responses. Kirschner and Bernardi (2007) and Bernardi and Kirschner (2008) proposed models which utilize both user’s responses and system’s responses. But, in all these approaches, no attempt was made to understand the structure of user-system interactions. *We believe that understanding the structure of user-system interactions is the key to identifying an effective model to track contextual information.*

Bertomeu et al. (2006) made an attempt to understand the structure of user-system interactions. Along the lines of their work, we aim to identify models which reflect the underlying structure of user-system interactions. We propose three models based on the way in which the contextual information is utilized in the user-system interactions. Contextual information can sometimes be found beyond the immediate preceding responses (antecedents) as discussed in (Bertomeu et al., 2006). The approach proposed in this paper was able to identify contextual information present in such responses. Further, it was also able to identify the contextual information present in more than one antecedent.

The remainder of this paper is organized as follows. In section 3, we more precisely define the problem and introduce our terminology and notation conventions. In section 4, we categorize the interactions between user and system. We model the user-system interactions in section 5. We propose a relationship schema among the responses (user and system responses) in section 6. In section 7, using these relations, we present a novel method to identify contextual information for one of the models proposed in section 5. Finally, we present our experimental results in section 8 and conclude in Section 9.

## 3 Problem

Responses by both user and system in a user-system interaction can be grouped into a set based on the information shared among them. Each individual group is called ‘local contextual group’ (LCG) and the corresponding information (i.e. information present in every user response of that group) maintained by it is called ‘local contextual information’ (LCI) or ‘contextual information’. Given a user response, first we need to identify the LCG to which it belongs and then use the corresponding LCI to interpret the user response.

The following notation is used throughout this paper:

$lc_i$  denotes the  $i^{\text{th}}$  LCG.

$u_{kl}$  denotes the  $k^{\text{th}}$  user response and there are  $l$  LCGs just before this response is given by the user.

$s_{kl}$  denotes the  $k^{\text{th}}$  system response and there are  $l$  LCGs just before this response is given by system.

For every user response  $u_{kl}$ , there will be a corresponding system response  $s_{kl}$ . We define the pair  $(u_{kl}, s_{kl})$  as a **dialogue unit**  $d_{kl}$ .

The user response  $u_{kl}$  can either belong to any of the previous LCGs  $lc_i = 1,2,3 \dots l$  or it can lead to the formation of new LCG  $lc_{l+1}$ . This is because the user can only either refer to the past information or can provide new information. User cannot refer to future local contexts (i.e.  $i > l+1$ ).

The system response  $s_{kl}$  can only belong to any of the previous local contexts  $lc_i = 1,2,3 \dots l$ . It cannot belong to  $lc_{l+1}$  or any of the other future local contexts. This is because the system can only provide output for the past ( $i \leq l$ ) user responses. Hence, only a user response can create a new LCG.

So there are two primary steps to identify contextual information of a user response  $u_{kl}$ : (a) To identify all the LCGs present in the interaction and (b) To find the corresponding LCG to which  $u_{kl}$  belongs.

#### 4 User-System Interactions

Kato et al. (2004) categorized the interactions between user and system into two types: Browsing type and Gathering type. In our experiments, we found a similar and more finer categorization to be helpful for analyzing the interactions:

**1) Strongly Coherent interaction:** In this kind of interaction, the user interacts with the system with a topic in mind and a goal to achieve. In our experiments, we found that most of the responses in such an interaction are closely related with each other (section 8).

**2) Coherent interaction:** In this kind of interaction, the user only knows about the topic and he does not have any specific goal. Here, the responses may not be as closely related as in strongly coherent interactions.

**3) Weakly Coherent interaction:** In this kind of interaction, the user neither has a topic nor a goal. Most of the responses in this type of interaction may not be related with each other.

#### 5 Modeling User-System Interaction

Depending on the way in which the contextual information can be utilized in the user-system interactions, we propose the following three models:

**1) Linear Disjoint Model:** In this model, the following three conditions hold true:

**condition 1:**  $u_{kl}$  can belong to only one LCG.

**condition 2:**  $u_{kl} \in lc_i$  or  $u_{kl} \in lc_{i+1}$ , where  $i = l$ .

This implies that user response can only either belong to the immediate previous LCG or it can form

a new LCG.

**condition 3:** All LCGs are disjoint.

This implies that responses belonging to a LCG can be interpreted without depending on the information present in responses belonging to any of the other LCGs.

For example, let us consider a Linear Disjoint interaction (i.e. user-system interaction which can be modeled by Linear Disjoint Model) shown in Figure 4. In this example,  $d_{10}$ ,  $d_{20}$  belong to first LCG and  $d_{31}$ ,  $d_{41}$  belong to second LCG. We can interpret the responses belonging to second LCG without depending on the information present in responses belonging to first LCG.

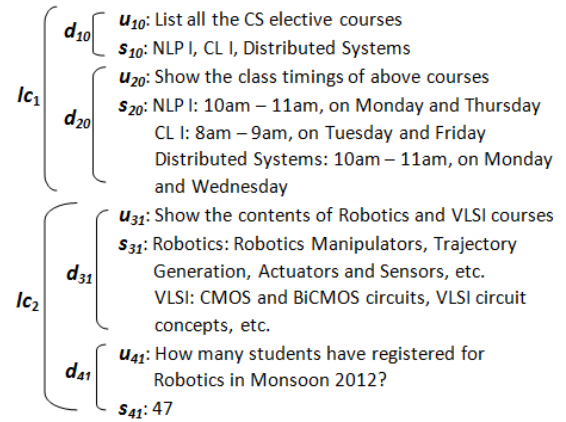


Figure 4: An example of Linear Disjoint interaction

**2) Linear Coincident Model:** In this model, condition 1 and condition 2 hold true. Condition 3 does not hold true if  $lc_i$  and  $lc_j$  are adjacent (i.e.  $|j-i| = 1$ ). This implies that interpreting responses belonging to a LCG may need the information present in the responses belonging to its adjacent LCG.

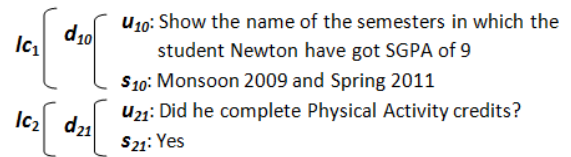


Figure 5: An example of Linear Coincident interaction

For example, let us consider an example of Linear Coincident interaction shown in Figure 5. In this example,  $d_{10}$  belong to first LCG and  $d_{21}$  belong to second LCG.  $u_{21}$  corefer the student name 'Newton' present in  $u_{10}$ . Hence interpreting the

responses belonging to second LCG needs the information present in responses belonging to first LCG.

It may also be noted that  $d_{10}$  and  $d_{21}$  may appear to belong to the same LCG but this is not so. If we assign both  $d_{10}$  and  $d_{21}$  to same LCG, then the information present in  $d_{10}$  would be used to interpret  $u_{21}$ . In this case  $u_{21}$  would be interpreted as ‘Did Newton complete Physical Activity credits in Monsoon 2009 and Spring 2011’. But the user’s intention is to know whether Newton completed Physical Activity credits or not (in any of the semesters). Hence both  $d_{10}$  and  $d_{21}$  cannot belong to same LCG.

**3) Non-Linear Model:** In this model, all the three conditions may or may not hold true. This implies that user response can belong to more than one LCG. Also interpreting responses belonging to a LCG may need the information present in the responses belonging to any of the other LCGs. Identification of contextual information in such interactions is very difficult compared to Linear Disjoint and Linear Coincident interactions. Complexity of contextual information present in various models is as follows:

*Linear Disjoint Model < Linear Coincident Model < Non-Linear Model*

## 6 Relationships between User Response and Dialogue Units

In a semantic template based non-contextual NLIDB system (Gupta et al., 2012), entities identified in semantic stage (explicit entities) are mapped onto the domain conceptual model based on an entity relationship graph (ER graph). A shortest path (sub-graph) in the ER graph connecting the explicit entities is computed. Implicit entities are the entities in the sub-graph which connect the explicit entities. For every user response, a sub-graph is generated. So for every dialogue unit, there exists a sub-graph.

Between user response ( $u_{kl}$ ) and dialogue units ( $d_{ij}$  where  $i < k$  and  $j < l$ ), we define the following relationships based on their corresponding sub-graphs:

(1) **Strong Link:**  $u_{kl}$  and  $d_{ij}$  are said to be strongly linked if their sub-graphs satisfy the following three properties.

*property 1:* there is at least one explicit entity ( $e_f$ ) in common.

*property 2:* there is at least one attribute ( $a_f$ ) of the

entity  $e_f$  in common.

*property 3:* there is at least one value ( $v_f$ ) to the attribute ( $a_f$ ) in common.

(2) **Link:**  $u_{kl}$  and  $d_{ij}$  are said to be linked if property 1, property 2 are satisfied and property 3 is not satisfied.

(3) **Weak Link:**  $u_{kl}$  and  $d_{ij}$  are said to be weakly linked if none of the properties are satisfied i.e. they either have implicit entities in common or no entity in common.

For example, let us consider the user-system interaction shown in Figure 6. Here  $u_{20}$  and  $d_{10}$  are strongly linked because they both have common explicit entity ‘course’, common attribute ‘course name’ and common value ‘Database Systems’ to that attribute. Similarly,  $u_{41}$  and  $d_{31}$  are strongly linked.  $u_{52}$  and  $d_{41}$  are linked because they only have the entity ‘professor’ in common.  $u_{31}$  and  $d_{20}$  are weakly linked because they don’t have any explicit entity in common.

## 7 Identifying contextual information in Linear Disjoint Model

To use contextual information in a user-system interaction, we need to perform two primary steps. First, we need to identify all the LCGs present in the interaction. Then, given a user response, we need to find the corresponding LCG to which it belongs. In our approach, we perform these two steps simultaneously.

In Linear Disjoint Model, a user response can either belong to the immediate previous LCG or it can form a new LCG. Let the user response be  $u_{kl}$ . That means there are already  $l$  LCGs before user has given this response. Now we need to find whether  $u_{kl}$  belongs to  $lc_l$  or not.

Suppose if  $u_{kl}$  is assigned to the LCG  $lc_l$ , the corresponding contextual information is used to interpret  $u_{kl}$ . Otherwise, a new LCG  $lc_{l+1}$  is created and  $u_{kl}$  is assigned to  $lc_{l+1}$ .

We use the relationships between user responses and dialogue units to determine whether  $u_{kl}$  belongs to  $lc_l$  or not. The intuition behind using these relationships is given below:

1) If  $u_{kl}$  is *strongly linked* to **any** dialogue unit belonging to  $lc_l$ , then it indicates that the user might be referring to the information present in  $lc_l$  and hence  $u_{kl}$  is assigned to  $lc_l$ .

2) If  $u_{kl}$  is *linked* to **any** dialogue unit belonging to  $lc_l$ , then it indicates that user might be reducing focus on the information present in  $lc_l$  and hence



the system creates a new LCG  $lc_{l+1}$  and assigns  $u_{kl}$  to  $lc_{l+1}$ . Since reducing focus may not always lead to formation of new LCG, system confirms with user by asking some questions.

3) If  $u_{kl}$  is *weakly linked* to **any** dialogue unit belonging to  $lc_l$ , then it indicates that user might not be referring to the information present in  $lc_l$  and hence the information present in  $lc_l$  is not used as contextual information. A new LCG  $lc_{l+1}$  is created and  $u_{kl}$  is assigned to  $lc_{l+1}$ .

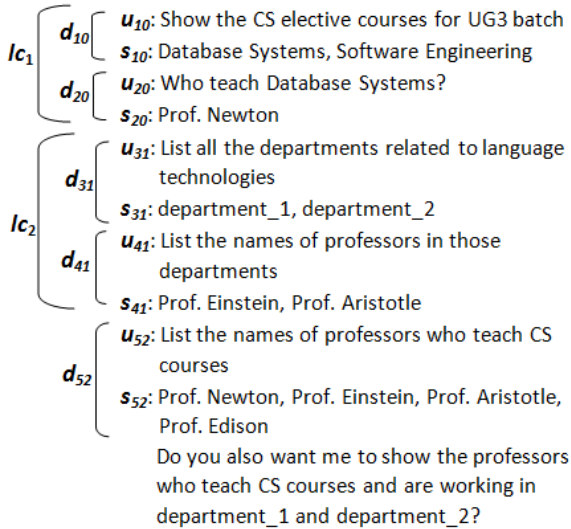


Figure 6: An example of Linear Disjoint model

For example, let us consider a Linear Disjoint interaction shown in Figure 6. Since  $u_{20}$  and  $d_{10}$  are strongly linked, we use the information present in  $d_{10}$  as the contextual information for  $u_{20}$ . Hence the output will be the names of professors who teach the course ‘Database Systems’ for UG3 batch.

As  $u_{31}$  and  $d_{20}$  are weakly linked, information present in  $d_{10}$  and  $d_{20}$  are not used as contextual information for  $u_{31}$ . Also a new LCG  $lc_2$  is created and  $u_{31}$  is assigned to  $lc_2$ . Similarly  $u_{41}$  uses information present in  $d_{31}$  as contextual information because they are strongly linked.

As  $u_{52}$  and  $d_{41}$  are linked, the user might be reducing focus on the information present in  $lc_2$ . Hence,  $u_{52}$  is interpreted without using the information present in  $lc_2$  as contextual information and later system confirms with user by asking some questions.

## 8 Experiments and Discussions

We carried out experiments on university related queries. Using the existing non-contextual NLIDB

system (Gupta et al., 2012), we have developed 110 dialogues which cover a wide range of topics such as course registration, seminar talks, credit requirements and cultural events. Each dialogue contains a sequence of user and system responses (or turns). On an average, each dialogue contains about 12 responses, corresponding to a total of 1320 responses.

Out of these 110 dialogues, 40 dialogues are of strongly coherent type, 40 dialogues are of coherent type and 30 dialogues are of weakly coherent type. We found that 96.6% of these dialogues belong to Linear Disjoint Model and 3.4% of the dialogues belong to Linear Coincident Model. We did not find any dialogues belonging to Non-Linear Model. This indicates that the method proposed in this paper is sufficient to identify contextual information in most of the real-time interactions.

	Strong Links	Links	Weak Links
Strongly Coherent interaction	72.84%	22.89%	4.29%
Coherent interaction	46.25%	43.75%	10%
Weakly Coherent interaction	34.34%	41.34%	24.34%

Table 1: Average percentage of relationships observed in different types of interactions

Table 1 shows the average percentage of various relationships (proposed in section 6) observed in different types of interactions. In a strongly coherent interaction type, higher percentage of strong links are observed. This is consistent with the definition of strongly coherent interaction. In such an interaction, user interacts with the system with a topic in mind and a goal to achieve. At each stage of the interaction, the user tries to move closer to the goal. Hence, we can expect the user to construct a query (or response) using the information obtained from the previous queries. This also explains the presence of a very small percentage of weak links.

From the definition of coherent interaction type, one would expect a higher percentage of links than strong links. On the contrary, we found almost equal percentage of strong links and links. This is because in a coherent interaction, the user can ask about various details regarding a topic. We can call these details as short term goals (or temporary

goals). In contrast to strongly coherent interaction where user has a single goal (long term goal) to achieve, coherent interaction contains many short term goals.

User may not get an answer for every short term goal in a single query. Hence, we can expect the user to ask multiple queries (but these are much less than the total number of queries used to achieve long term goal) to achieve short term goals. The interaction corresponding to every short term goal have high percentage of strong links than links. Interactions corresponding to every two short term goals are expected to connect with either links or weak links. But since we have a fixed topic, we can expect higher probability for links to connect those short term goals. As there can be many short term goals, percentage of links will be also high.

In a weakly coherent interaction, higher percentage of weak links are observed compared to other two types of interactions. This is because the user neither has topic nor a goal to achieve. Hence, while interacting with the system, the user may randomly pick topics and ask various details related to those topics. Once a topic is chosen, the interaction can be viewed as a coherent interaction. Hence, we can see almost the same percentage of strong links and weak links. Notice that there is a higher probability for interactions with different topics to be weakly linked with each other. As a user may frequently change the topics, we can see the increase in the percentage of weak links.

Table 2 shows the average number of local contexts, average length of local context (i.e. total number of responses in each local context) observed in different types of interactions. As discussed earlier, in a strongly coherent interaction, the user has a fixed and a single goal to achieve. So, we can expect most of the queries to be related to each other. Hence, this type of interactions contain less number of local contexts and each local context has more responses.

Coherent interactions contain many short term goals and each short term goal is expected to contain less number of responses compared to the long term goals present in strongly coherent interactions. So this type of interactions contain comparatively more number of local contexts and smaller average length than strongly coherent interactions.

In weakly coherent interactions, user can change the topics very often and hence contain higher number of local contexts and least average length.

	Number	Length
Strongly Coherent interaction	2.2	4.67
Coherent interaction	3	1.88
Weakly Coherent interaction	3.83	1.43

Table 2: Average number and average length of local contexts observed in different types of interactions

We applied the method proposed in section 7 to 106 Linear Disjoint dialogues (which constitute 96.6% of the total dialogues). The results obtained are impressive. For each dialogue, we evaluated the percentage of the queries for which the corresponding contextual information has been identified correctly. The contextual information is identified with 100% accuracy for 78 dialogues i.e. our method successfully identified the appropriate context for every user response of those dialogues. The contextual information for 13 dialogues has been identified with 10 to 20% error. 9 dialogues are found with error greater than 40%.

## 9 Conclusion

In this paper we categorized user-system interactions and then proposed three models (Linear Disjoint Model, Linear Coincident Model and Non-Linear Model) depending on the way in which the contextual information can be utilized in the interactions. We proposed a new relationship schema among the responses. Central in our approach is the use of these relationships to identify contextual information in Linear Disjoint interactions. Furthermore, we evaluated our approach on university related queries and the results confirm the viability of the proposed approach. In our corpus, we found that 96.6% of the total interactions are Linear Disjoint interactions. Hence the method proposed in this paper is sufficient to identify contextual information in most of the real-time interactions.

In the future, we plan to investigate how to identify the model of an interaction. We also intend to identify contextual information in Linear Coincident interactions and Non-Linear interactions.

## References

- Ioannis Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. 1995. Natural language interfaces to databases-an introduction. *arXiv preprint cmp-lg/9503016*.
- Raffaella Bernardi and Manuel Kirschner. 2008. Context modeling for iqa: the role of tasks and entities. In *Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions*, pages 25–32. Association for Computational Linguistics.
- Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. Contextual phenomena and thematic relations in database qa dialogues: results from a wizard-of-oz experiment. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8. Association for Computational Linguistics.
- Joyce Y Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, pages 23–30.
- Alessandra Giordani and Alessandro Moschitti. 2009. Syntactic structural kernels for natural language interfaces to databases. In *Machine Learning and Knowledge Discovery in Databases*, pages 391–406. Springer.
- Alessandra Giordani. 2008. Mapping natural language into sql in a nlib. In *Natural Language and Information Systems*, pages 367–371. Springer.
- Abhijeet Gupta, Arjun Akula, Deepak Malladi, Puneeth Kukkadapu, Vinay Ainavolu, and Rajeev Sangal. 2012. A novel approach towards building a portable nlib system using the computational paninian grammar framework. In *Asian Language Processing (IALP), 2012 International Conference on*, pages 93–96. IEEE.
- Tsuneaki Kato, Junichi Fukumoto, and Fumito Masui. 2004. Question answering challenge for information access dialogue-overview of ntcir-4 qac2 sub-task 3. In *Proceedings of the 5th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 291–297.
- Manuel Kirschner and Raffaella Bernardi. 2007. An empirical view on iqa follow-up questions. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium*.
- Manuel Kirschner and Raffaella Bernardi. 2010. Towards an empirically motivated typology of follow-up questions: the role of dialogue context. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 322–331. Association for Computational Linguistics.
- Yunyao Li, Huahai Yang, and HV Jagadish. 2005. Nalix: an interactive natural language interface for querying xml. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 900–902. ACM.
- Xiaofeng Meng and Shan Wang. 2001. Nchiql: The chinese natural language interface to databases. In *Database and Expert Systems Applications*, pages 145–154. Springer.
- Matteo Negri and Milen Kouylekov. 2007. who are we talking about? tracking the referent in a question answering series. In *Anaphora: Analysis, Algorithms and Applications*, pages 167–178. Springer.
- Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 149–157. ACM.
- Niculae Stratica, Leila Kosseim, and Bipin C Desai. 2005. Using semantic templates for a natural language interface to the cindi virtual library. *Data & Knowledge Engineering*, 55(1):4–19.
- Mingyu Sun and Joyce Y Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems*, 20(6):511–526.

# Iterative Development and Evaluation of a Social Conversational Agent

Annika Silvervarg, Arne Jönsson

Department of Computer and Information Science

Linköping University, Linköping, Sweden

annika.silvervarg@liu.se, arne.jonsson@liu.se

## Abstract

We show that an agent with fairly good social conversational abilities can be built based on a limited number of topics and dialogue strategies if it is tailored to its intended users through a high degree of user involvement during an iterative development process. The technology used is pattern matching of question-answer pairs, coupled with strategies to handle: follow-up questions, utterances not understood, abusive utterances, repetitive utterances, and initiation of new topics.

## Introduction

Social aspects of conversations with agents, such as small talk and narrative storytelling, can have a positive effect on peoples general interest in interacting with it and help build rapport (Bickmore, 2003). It can also be utilised to develop a relationship and establishing trust or the expertise of the agent (Bickmore and Cassell, 1999). We are interested in exploring if and how these and other effects transfer to an educational setting where children and teenagers interact with pedagogical agents in virtual learning environments. We see several reasons to incorporate social conversation with such agents, for example, it allows for cognitive rest, it can increase overall engagement and receptivity and it can make students feel more at ease with a learning task or topic (Silvervarg et al., 2010). There has, however, been few attempts to understand the users' behaviour in social conversations with pedagogical agents (Veletsianos and Russell, 2013) and embodied conversational agents (Robinson et al., 2008).

In this paper we report on how we iteratively have worked with addressing the questions of 1) what do users talk about during social conversation with a pedagogical agent, 2) how do users

talk during social conversation with a pedagogical agent, 3) how does the answers to 1) and 2) affect the dialogue functions needed to implement social conversation with a pedagogical agent.

## A social conversational pedagogical agent

Our work extends a virtual learning environment with an educational math game named "The Squares Family" (Pareto et al., 2009). A crucial part of the environment is a pedagogical agent, or more specifically a teachable agent (Biswas et al., 2001). While the student is playing the game, the agent "learns" the rules of the game in two ways, by observation or through on-task multiple choice questions answered by the user. A teachable agent is independent and can act on its own, yet is dependent on the student to learn rules and strategies. The intended users are 12-14-year-old students, and the teachable agent is designed as having the same age or slightly younger.

The conversational module for off-task conversations has been developed as a rather independent module of the learning environment. Off-task conversation is based on a character description of the teachable agent that is consistent with the overall role of the agent as a peer in the environment.

The challenge can be seen as a question of managing the students' expectations on the agent's abilities. Our approach was to frame and guide the interaction with the student in such a way that, ideally, the shortcomings and knowledge gaps of the agent never become a critical issue for achieving a satisfying communication. We have therefore chosen to work with user-centred agile system development methods to be able to capture the users' behaviour and tailor the agent's conversational capabilities to meet their expectations. This includes combining focus group interviews and Wizard-of-Oz role-play (Dahlbäck et al., 1993) with development and evaluation of prototypes, surveys and analyses of natural language interaction logs.

The off-task conversation is implemented using a slightly extended version of AIML, Artificial Intelligence Markup Language (Wallace, 2010). AIML works on the surface level and map user utterances to system responses. User utterances can consist of words, which in turn consist of letters, numerals, and the wildcards `_` and `*`, which function like words. Synonyms are handled using substitutions and grammatical variants through several different patterns for the same type of question and topic.

Responses consist in their simplest form of only plain text. It is also possible to set or get data in variables and predicates, give conditional responses, choose a random response from a set of responses, and combinations of these. AIML also allows for handling a limited context by either referring to the systems last utterance or a topic that span multiple exchanges.

### Prototype 1

In the first iteration an agent persona was developed through focus groups with 20 target users. The persona sketch formed the basis for WOZ-style role play, in which students simulated off-task conversations in the game. Three students played the part of the agent, and four students played the role of the user. The resulting 12 dialogues were analysed according to topics, linguistic style and dialogue phenomenon. A number of new topics emerged that had not been brought up in the focus groups. The linguistic style of the utterances could be characterised as grammatical, short sentences, with the use of smileys and "chat-expressions". The dialogue mostly consisted of unconnected question and answer pairs, but some instances of connected dialogue with 3-4 turns occurred. The initiative was evenly distributed between user and system. There were frequent use of elliptical expressions, mostly questions of the type "what about you", but no anaphora.

Based on these findings the first prototype implemented basic question-answer pairs, a strategy for follow-up questions from the agent and user, a topic model with 6 topics that could be initiated by the agent (to allow for mixed-initiative dialogue), and a very simple strategy to handle utterances that the agent could not understand. To handle variations in user input (choice of words and grammatical variations) the system used substitutions where, for example, synonyms and hyponyms

were substituted for a "normalised" word, and variations of patterns that used the "normalised" keywords. The agent's replies were sometimes randomly chosen from a set of 3-5 variants to get some variation if the user asked the same question several times. Follow-up questions were randomly attached to half of the answers the agent gave to questions from the user. When the agent did not understand a user utterance it said so three out of four times, but in one out of four it instead initiated a new topic and posed a question to the user. The agent also initiated a new topic when the user gave an acknowledgement, such as ok, after an answer from the agent.

To evaluate the system a total of 27 students tested the prototype. After a short introduction to the project and the system they played the game for 10 minutes, chatted with the agent for 5 minutes, then played the game for 5 minutes and chatted for 5 minutes again. Analysis of the corpus showed that failed interpretations had to be dealt with. Many of the failed interpretations were due to linguistic variations on known topics, and most of all acknowledgments, but also greetings and follow-up questions. Topics also needed to be expanded, both new topics, for example age, food, pets, favourite colour and cars, but also more sub-topics related to, for example, computer games, school subjects, and TV. Topics initiated by the agent were proved to be a good strategy and implied that the model should be expanded with new topics and more sub-topics. If the agent could initiate more topics it would both make the dialogue more fun but also help the user to know what type of topics the agent could talk about. A recovery strategy to deal with utterances that the system still were not able to interpret correctly, for example utterances out of scope, was also needed to help the user understand what they could talk about and avoid losing track of the conversation. The recovery strategy should also include ways of breaking long sequences of utterances that included, for example, random letters and "Ok" sub-dialogues, and steer the dialogue back to a topic.

Evaluation of the prototype also showed that hazing, testing and flaming occurred rather frequently, for example comments and questions regarding the agent's appearance, intellectual capacities, and sexual orientation and activities. To fend this off, the agent also needed to recognise such utterances and have a strategy to direct the dialogue

in a more productive direction.

## Prototype 2

The second prototype expanded the agent's dialogue functionality with a recovery strategy for utterances not understood by the agent, a strategy to handle abusive utterances, a strategy for delayed follow-up questions by the user (e.g. Agent: How are u? User: Fine. Agent: Ok. User: wbu?), an extended topic model, a better model for agent initiative and flow of the dialogue, and extended vocabulary.

The recovery strategy for utterances not understood by the agent was implemented similar to the one used by SGT Star (Artstein et al., 2009); first a simple "clarification", utterance 2 in Figure 1, then, a "stall", followed by a "move on", and finally a "prompt" that directs the user to a new topic. In our system the agent stall is a general prompt for the user to initiate a new topic, utterance 4 in Figure 1, if this fails the agent proposes a topic, utterance 6 in Figure 1, and as a last resort takes the initiative and asks a question, utterance 8 in Figure 1.

- 1 User: fgd
- 2 Agent: I did not understand that
- 3 User: cb
- 4 Agent: I did not catch that, can we talk about something else?
- 5 User: s
- 6 Agent: Let's talk about school
- 7 User: s
- 8 Agent: My school is okay, a little too much homework, wbu?

Figure 1: Example of how the agent handles utterances that are not understood.

The strategy to handle abusive utterances are similar in that the first response is general, such as "eh, let's talk about something else", followed by two attempts to change the topic, randomly done either by a general request for the user to suggest a topic or by the agent to introduce a new topic, followed by a remark that further abuse will result in a report to the teacher. If the user continued with abusive utterances the loop starts again

To avoid repetitive utterance sequences such as many greetings, laughs or acknowledgements in a row, the agent initiated new topics when those types of utterances were repeated. The AIML parameter topic was used to handle delayed follow-up questions from the user. The topic model used for this purpose was extended to include a total of

15 topics, where some were related, for example music and favourite artist.

Evaluation of the prototype was conducted in the same way as for prototype 1. This time with 42 users, 22 girls and 20 boys. Analysis of the chat logs revealed that the model for follow-up questions needed to be revised. Since follow-up questions were initiated randomly by the agent it sometimes asked for information the user already had provided, which seemed to irritate the user. The model for topic initiation by the agent could also be refined to provide more coherence in the dialogue. Another problem detected was that the strategy to use the current topic as context to interpret generic follow-up questions sometimes was overused and led to misunderstandings when the user tried to introduce a new topic. The agent thus needed a more sophisticated strategy to handle topics and topic shifts.

## Prototype 3

The main improvements of prototype 3 was the introduction of mini narratives, an improved strategy for follow-up questions and an improved strategy to introduce and continue a topic. For three main topics, free time, music and school, sub-topics were added, and when the agent took the initiative it tried to stay within topic and either tell a mini-narrative or introduce a sub-topic. Follow-up questions were now only posed if the user had not already provided answers to the question earlier in the conversation.

The conversational agent was evaluated at a Swedish School, where 19 students, from three classes, 12-14 years old, used the learning environment with the conversational agent during three lectures. The students played the game for about a total of 120 minutes and after every second game session a break was offered. During the first three breaks the students had to chat with the agent until the break ended, after that chatting was optional.

Table 1 shows the proportion of different types of user utterances in the logged conversations. The coding scheme is based on the coding schemes used by Robinson et al. (2010) to evaluate virtual humans. As can be seen in Table 1 most user utterances are "appropriate" in that they are either Information requests (Q), Answers (A), General dialogue functions (D) or Statements (S), but a total of 22% are "inappropriate", i.e. Incomprehensible

(G) or Abusive (H).

Table 1: Dialogue action codes and proportion of different agent utterances.

Code	Description	Prop
D	General dialogue functions, e.g. Greeting, Closing, Politeness	14%
H	Hazing, Testing, Flaming, e.g. Abusive comments and questions	11%
Q	Information Request, e.g. Questions to the agent	31%
R	Requests, e.g. Comments or questions that express that the user wants help or clarification	0%
A	Answer to agent utterances	18%
S	Statements	16%
G	Incomprehensible, e.g. Random key strokes or empty utterances	11%

As for the agent's responses it seems that the system handles most utterances appropriately although many of these are examples of requests for repair, see Table 2. The highest value 3, i.e. appropriate response, means that the agent understood the user and responded correctly. Request Repair, is when the system does not understand and asks for a clarification or request that the user changes topic. Partially appropriate, code 2, is typically used when the user's utterance is not understood by the agent, and the agent's response is to initiate a new topic. Inappropriate response, code 1, is when the system responds erroneously, typically because it has misinterpreted the user's utterance.

Table 2: Agent response codes and proportion of different agent responses.

Code	Description	Prop
3	Appropriate response	51%
2	Partially appropriate	15%
RR	Request Repair	30%
1	Inappropriate response	4%

Given a definition of Correct Response as any response that is not inappropriate, code 1 in Table 2, we see that prototype 3 handles 96% of the user's utterances appropriately or partly appropriate. The proportion of responses where the system correctly interprets the user's utterance is, however, only 54%, and there are still 11% Flaming/Hazing which also affects the number of repetitions, which is very high. Most of the not correctly interpreted utterances and the repetitions, occurs when the student is hazing/flaming or testing the system, e.g. none of the user's utterances

in Figure 1 is correctly interpreted (code 3) but all are correctly responded to (code 2).

## Prototype 4

The evaluation of prototype 3 did not indicate any need for more sophisticated dialogue functions but rather that the number of correctly interpreted utterances needed to increase. Therefore the focus of prototype 4 was to add and refine patterns used for interpretation of user utterances, for example adding more synonyms and expressions. It also included adding answers to some questions related to the already present topics, for example, questions on the agent's last name and questions and comments about game play. Since prototype 3 still had problems with a lot of abusive comments prototype 4 also included a revised strategy to handle abusive utterances, where the agent gradually tries to change the topic and finally stops responding if the abuse continues to long. If and when the user changes topic the strategy is reset.

The evaluation of prototype 4 comprise conversations with 44 students, 12-14 years old. The students used the system more than once which gives us 149 conversations with a total of 4007 utterances of which 2003 are from the agent. Each utterance was tagged with information about, dialogue function, topic, initiative, agent interpretation, agent appropriate response, and abuse.

Many of the utterances that the agent could not correctly interpret in prototype 3 were due to the fact that users did not engage in the conversation and did not cooperate, rather they were testing the agent, abusing it or just writing nonsense. We believe that the strategies we have developed to handle such utterances are more or less as good as a human. For the evaluation of prototype 4 we therefore modified the criteria for tagging an utterance as appropriate. An utterance was only appropriate if the agent responded as good as a human, taking into account that if a user utterance is very strange, a human cannot provide a very good answer either, see Table 3. In this new coding scheme we also removed the previous category RR where utterance that request repairs falls into R3 (if a human could not interpret the user utterance neither) or R2 depending on how appropriate they are in the context.

There are also cases when the agent's response may have been better if it was a human, but where it is not obvious how, or even that a human could

Table 3: Agent response values.

Code	Value
R3	Agent responses that a human could not have done better
R2	Agent responses that are ok but a human may have responded better
R1	Agent responses that are erroneous because the agent did not understand the student or misunderstood

do better. We tag these R2 as well not to give credit to the agent for such responses.

Table 4 shows topics with information on how many utterances in total belonged to each topic, and how well the agent responded to utterances within each topic (R1, R2 or R3), as well as the proportion of not appropriate or only partially appropriate responses in percentage. NO TOPIC is for utterances like greetings, requests for repair, random letters or words, and abuse. As can be seen in Table 4 the agent gives appropriate responses (R3) to 1399, i.e. 70%, of the users' utterances. Table 4 lists all the topics present in the corpus and shows that although given the opportunity to talk about anything, users tend to stick to a small number of topics.

Table 4: Topics present in the corpus and the number of appropriate responses (R3), partially appropriate response (R2), and non-appropriate responses (R1).

TOPIC	Tot	R3	R2	R1	Prop R1 + R2
NO TOPIC	534	527	50	6	10%
PERSINFO	317	189	147	32	56%
MUSIC	267	199	43	25	25%
SCHOOL	201	136	48	17	32%
FREE-TIME	177	136	35	6	23%
MATH-GAME	122	43	74	5	65%
COMP-GAME	103	80	19	4	22%
FOOD	38	15	18	5	61%
FAMILY	34	15	17	2	56%
FRIENDS	30	8	20	2	73%
MOVIES	24	19	3	2	21%
SPORT	21	12	6	3	43%
MATH	20	13	7	0	35%
ALCOHOL	5	2	3	0	60%
BOOKS	2	0	1	1	100%
CLOTHES	2	0	2	0	100%
FACE-BOOK	2	2	0	0	0%
PET	2	2	0	0	0%
TV	2	1	1	0	50%
<b>Total</b>	<b>2003</b>	<b>1399</b>	<b>494</b>	<b>110</b>	<b>30%</b>

To further investigate the utterances causing problems we looked at the responses tagged as R1

and R2 and classified them as caused by greetings (GREETING), questions (QUESTION), statements (STATEMENT) or utterances where correct interpretation depends on the dialogue history (HISTORY). The proportions of problematic utterances and the dialogue functions of these utterances are shown in Table 5.

Over half of the problematic utterances are questions. Of these the majority are regular questions, while 30% of them are specific follow up questions on a previously introduced topic. A small number are generic follow up questions either directly following an answer to a question posed by the agent (Agent: Do you like school?, User: yes, wbu?), or a free standing delayed question (Agent: Do you like school? User: Yes. Agent: ok, User: wbu?). Statements are causing 29% of the not appropriate answers, mainly statements and answers to questions. There are also some abusive comments and random utterances. Problems related to the dialogue history is comparatively small. It includes both answers, statements and different kinds of questions. Examples of utterances the agent cannot handle well are follow up questions on topics previously introduced by the agent or the user, statements that comment on previous answers, use of anaphora referring to previous questions or answers, users' attempt to repair when the agent does not understand, and delayed answers to questions asked more than one utterance before.

From Table 6 we see that most of the problems relate to a small number of topics. PERSINFO, FREETIME and MATHGAME have mainly problems with statements and questions. The agent has for example insufficient knowledge and ability to talk about the math game itself. It also lacks knowledge about personal information such as hair colour, eye colour and other personal attributes. MUSIC and SCHOOL are common topics where the user often tries to make follow up topics that the agent cannot handle.

## Conclusions

We have worked iteratively with user centred methods and rather straightforward natural language processing techniques to develop a social conversational module for a pedagogical agent aimed at students aged 12-14 years. The importance of involving students in the development process cannot be underestimated. Initially they



Table 5: Type of utterances that causes not appropriate responses, and their dialogue function.

	R1	R2	Tot	Prop
<b>GREETING</b>	<b>2</b>	<b>15</b>	<b>17</b>	<b>2,8%</b>
Greetings	2	15	17	2,8%
<b>QUESTION</b>	<b>72</b>	<b>244</b>	<b>316</b>	<b>52,6%</b>
Questions	45	141	186	30,9%
Specific Follow up Questions	21	74	95	15,8%
Generic Follow up Questions	1	17	18	3,0%
Answer + GFQ	1	8	9	1,5%
Abuse	2	5	7	1,2%
<b>STATEMENT</b>	<b>17</b>	<b>158</b>	<b>175</b>	<b>29,1%</b>
Statement	9	71	80	13,3%
Answer	4	45	49	8,2%
Acknowledgement	2	16	18	3,0%
Abuse	1	16	17	2,8%
Random	1	8	9	1,5%
<b>HISTORY</b>	<b>19</b>	<b>74</b>	<b>93</b>	<b>15,5%</b>
Answer	3	25	28	4,7%
Statement	2	26	28	4,7%
SFQ	4	10	14	2,3%
Random	8	3	11	1,8%
GFQ	1	4	5	0,8%
Question	1	4	5	0,8%
Acknowledgment		2	2	0,3%

Table 6: The distribution of different types (G: Greetings, H: History, Q: Questions, S: Statements) of problematic utterance for different topics.

TOPIC	G	H	S	Q	Tot
PERSINFO		11	38	130	179
MATHGAME		9	23	47	79
MUSIC		17	30	21	68
SCHOOL		23	21	21	65
NO TOPIC	19	10	21	6	56
FREETIME		6	13	22	41
COMPGAME		2	8	13	23
FOOD		1	7	15	23
FRIENDS		1	1	20	22
FAMILY		1	7	11	19
SPORT			5	4	9
MATH		2	1	4	7
MOVIES			2	3	5
ALCOHOL			1	2	3
CLOTHES				2	2
BOOKS				2	2
TV				1	1
<b>Total</b>	<b>19</b>	<b>83</b>	<b>178</b>	<b>324</b>	<b>604</b>

gave us valuable insights on the capabilities of an agent capable of social conversation. In the iterations to follow they provided feedback on how to refine the conversation to handle both "normal" conversation as well as not so conventional conversation. Using questionnaires to measure system performance or as an instrument for further development is not fruitful (Silvervarg and Jönsson, 2011). We have instead relied on analysis of the

logs to find bugs, and detect patterns that suggest lack or sophistication of dialogue capabilities that should be added or refined.

The strategy has been fairly successful. We seems to have captured what users talk about very well. The number of topics is surprisingly small given that the user can introduce any topic they want. A possible improvement could be to include a more elaborate model for topics and subtopics for some topics. There are also still knowledge gaps concerning some questions within topics, such as personal attributes and traits of the agent.

How they talk about the topics are also fairly well understood, in that the dialogue capabilities needed have been discovered and implemented. It may be that addition of anaphora resolution could improve the agents responses, but that would probably be a marginal improvement, since problems related to anaphora are very rare. Some of the problems are related to the large variation of how the same question or statement can be expressed, and the limited power of interpretation based on keywords, but this does not seem to be a big problem. The same can be said for spelling mistakes. Inclusion of an automatic spellchecker may increase the successful interpretations, but probably only to a small degree.

A remaining problem that is hard to address is the fact that some users are very uncooperative. They deliberately test the system or are just not engaging in the dialogue but rather write nonsense or abuse. Previous studies have shown that there seem to be three types of users (Silvervarg and Jönsson, 2011): 1) those that really try to use the system and often also like it, 2) users that do not use the system as intended, but instead tries to find its borders, or are bored and never tries to achieve an interesting dialogue, but rather resorts to flaming/testing/hazing, and 3) those that are in between. Users of type 1 are rather unproblematic, as long as the agent has enough topics and sub-topics they will have a meaningful conversation. Users of type 2 will probably never be engaged in a meaningful conversation with the agent no matter how sophisticated it is. Focus must instead be to avoid users of type 3 to adhere to type 2 behaviour, which could be achieved by having a variety of techniques to handle abusive and testing behaviour and enough topics and sub-topics to allow for a varied enough conversation, as presented in this paper.

## References

- Ron Artstein, Sudeep Gandhe, Jillian Gerten, Anton Leuski, and David Traum. 2009. Semi-formal evaluation of conversational characters. *Languages: From Formal to Natural*, pages 22–35.
- T Bickmore and J. Cassell. 1999. Small talk and conversational storytelling in embodied interface agents. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*.
- T. Bickmore. 2003. *Relational Agents: Effecting Change through Human-Computer Relationships*. Ph.D. thesis, Media Arts & Sciences, Massachusetts Institute of Technology.
- G. Biswas, T. Katzlberger, J. Brandford, Schwartz D., and TAG-V. 2001. Extending intelligent learning environments with teachable agents to enhance learning. In J.D. Moore, C.L. Redfield, and W.L. Johnson, editors, *Artificial Intelligence in Education*, pages 389–397. Amsterdam: IOS Press.
- Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of oz studies – why and how. *Knowledge-Based Systems*, 6(4):258–266. Also in: *Readings in Intelligent User Interfaces*, Mark Maybury & Wolfgang Wahlster (eds), Morgan Kaufmann, 1998.
- Lena Pareto, Daniel L. Schwartz, and Lars Svensson. 2009. Learning by guiding a teachable agent to play an educational game. In *Proceedings of AIED*, pages 662–664.
- Susan Robinson, David Traum, Midhun Ittycheriah, and Joe Henderer. 2008. What would you ask a conversational agent? observations of human-agent dialogues in a museum setting. In *Proceedings of LREC 2008*, Jan.
- Susan Robinson, Antonio Roque, and David R. Traum. 2010. Dialogues in context: An objective user-oriented evaluation approach for virtual human dialogue. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.
- Annika Silvervarg and Arne Jönsson. 2011. Subjective and objective evaluation of conversational agents in learning environments for young teenagers. In *Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.
- Annika Silvervarg, Agneta Gulz, and Björn Sjäodén. 2010. Design for off-task interaction – rethinking pedagogy in technology enhanced learning. In *Proceedings of the 10th IEEE Int. Conf. on Advanced Learning Technologies, Tunisia*.
- George Veletsianos and Gregory Russell. 2013. What do learners and pedagogical agents discuss when given opportunities for open-ended dialogue. *Journal of Educational Computing Research*, 48(3):381–401.
- Richard S. Wallace. 2010. Artificial intelligence markup language. URL:<http://www.alicebot.org/documentation/>.

# A Hybrid Morphological Disambiguation System for Turkish

**Mucahid Kutlu**

Dept. of Computer Science & Eng.,  
Ohio State University, Ohio, USA  
kutlu@cse.ohio-state.edu

**Ilyas Cicekli**

Department of Computer Engineering,  
Hacettepe University, Ankara, Turkey  
ilyas@cs.hacettepe.edu.tr

## Abstract

In this paper, we propose a morphological disambiguation method for Turkish, which is an agglutinative language. We use a hybrid method, which combines statistical information with handcrafted rules and learned rules. Five different steps are applied for disambiguation. In the first step, the most likely tags of words are selected. In the second step, we use handcrafted rules to constrain possible parses or select the correct parse. Next, the most likely tags are selected for still ambiguous words according to the suffixes of the words that are unseen in the training corpus. Then, we use transformation-based rules that are learned by a variation of Brill tagger. If the word is still ambiguous, we use some heuristics for the disambiguation. We constructed a hand-tagged dataset for training and applied a ten-fold cross validation with this dataset. We obtained 93.4% accuracy on the average when whole morphological parses are considered in calculation. The accuracy increased to 94.1% when only part-of-speech tags and inflections of last derivations are considered. Our accuracy is 96.9% in terms of part-of-speech tagging.

## 1 Introduction

Digital text sources increase every day and people can reach digital sources via Internet. The automatic processing of these sources becomes crucial in order to use and manage them effectively. Many NLP researchers work on different topics like summarization of texts, translation between natural languages, information extraction, etc. However, working on natural languages has difficulties due to their ambiguous natures. Since constraining possible morphological parses of words with disambiguation methods reduces the ambiguity problem, morphological disambiguation is crucial for performing better operations on texts.

In this paper, we propose a method for morphological disambiguation of Turkish texts.

Turkish is an agglutinative language. Agglutinative languages generate words by joining affixes together and each affix represents one unit of meaning. This property loads many meanings to a single word. Since suffixes can increase the ambiguity by generating totally different words, the morphological analysis of words with many suffixes is much harder.

Our hybrid system uses statistical knowledge, learned transformation-based rules, handcrafted rules and some heuristics in the disambiguation process. In our system, we first obtain the statistical information about words and suffixes like frequencies of their corresponding morphological parses and learn transformation-based rules. In disambiguation, we use an iterative approach that applies techniques from the most reliable technique to the less reliable technique. First, if the word exists in our training set, we select the morphological parse with the highest frequency. Then, we consider handcrafted rules of SupervisedTagger software (Daybelge and Cicekli, 2007) for disambiguation. Next, we use statistical information about suffixes and select the morphological parse with the highest frequency in the corpus for that suffix. Then, we apply the disambiguation rules learned from the corpus. Finally, we use some heuristics, which depend on some statistical information to select morphological parses for still ambiguous words.

The major contribution of this study is our hybrid morphological system, which combines the statistical, and rule based approaches for disambiguation. The accuracy of our hybrid disambiguation system is quite good when examined with the Turkish language which has a very flexible set of grammar rules and very ambiguous words. Our combined approach works very well even with less statistical language sources such as a small hand-tagged corpus. Although the size of our hand-tagged corpus is relatively small, the performance of our hybrid system is very good. The performance of the presented hybrid system

can be improved further when a bigger hand-tagged corpus is available.

The rest of the paper consists of four sections. Section 2 describes the related work in morphological disambiguation. Section 3 explains the proposed system. Section 4 describes the corpus and presents the performance results of the system. Section 5 concludes the paper by summarizing the study and discusses some future work.

## 2 Related Work

The related work about morphological disambiguation can be divided into three categories: statistical, rule based and hybrid which is the combination of the two approaches. Statistical approaches select the morphological parses using a probabilistic model that is built with the training set consisting of unambiguously tagged texts. There are various models used in the literature, such as, maximum entropy models (Ratnaparkhi, 1996; Toutanova and Manning, 2000), Markov Model (Church, 1988) and hidden Markov Model (Cutting et. al., 1992). In rule based methods, hand crafted rules are applied in order to eliminate some incorrect morphological parses or select correct parses (Daybelge and Cicekli, 2007; Oflazer and Tür, 1997; Voutilainen, 1995). These rules can also be learned from a training set using a transformation based (Brill, 1995) or memory based (Daelemans, 1996) learning approaches. There are also studies that combine statistical knowledge and rule based approaches (Leech et al., 1994; Tapanainen and Voutilainen, 1994; Oflazer and Tür, 1996).

The disambiguation studies can also be divided according to the languages they are applied. Levinger et al. (1995) used morpho-lexical probabilities learned from an untagged corpus for morphological disambiguation of Hebrew texts. Hajic and Hladká (1998) used maximum entropy modeling for Czech, which is an inflectional language. Morphological disambiguation of agglutinative languages, such as Turkish, Hungarian, Basque, etc., is harder than others because they have more morphological parses per words. Megyesi (1999) has used Brill's POS tagger with extended lexical templates to Hungarian. Hajic (2000) extended his work for Czech to five other languages including Hungarian. Ezeiza et al. (1998) combined statistical and rule based disambiguation methods for Basque. Rule based methods (Oflazer and Tür, 1997; Daybelge and Cicekli, 2007) and trigram-based statistical model (Tür et al., 2002) are used for the disambigua-

tion of Turkish words. Yüret and Türe (2006) propose a decision list induction algorithm for learning morphological disambiguation rules for Turkish. Sak et al. (2007) apply perception algorithm in disambiguation of Turkish Texts.

## 3 Disambiguation System

A Turkish word can have many morphological parses containing many morphemes that give us morphological information about the word. For example, the word "çiçekçi" (florist) has the following *morphological parse* (MP)<sup>1</sup>:

$$\begin{aligned} & \text{çiçek+Noun+A3sg+Pnon+Nom} \\ & \quad \wedge \text{DB+Noun+Agt+A3sg+Pnon+Nom.} \end{aligned} \quad (1)$$

The first part gives us the stem, which is "çiçek" (flower). We define the rest of the parse as the *whole tag* of the word. In parse, " $\wedge$ DB" shows that the word is derived from one type to another and its meaning has changed rather than its inflection. We define the final morphemes after the last derivation as the *final tag* of the word. For this example, the whole tag is:

$$\begin{aligned} & \text{Noun+A3sg+Pnon+Nom} \\ & \quad \wedge \text{DB+Noun+Agt+A3sg+Pnon+Nom,} \end{aligned} \quad (2)$$

and the final tag is:

$$\text{Noun+A3sg+Pnon+Nom}$$

where the type of derivation "Agt" is ignored. The rules that are learned by our system depend on morphological parses, whole tags or final tags of words.

Our disambiguation system consists of two main parts: *training* and *disambiguation*. The training corpus is used for the induction of disambiguation rules and the generation of the tables *Most Likely Tag of Word Table* (WordTbl) and *Most Likely Tag of Suffix Table* (SuffixTbl). WordTbl is used to retag the corpus by our Brill tagger in order to learn rules. WordTbl, SuffixTbl and the learned rules are used in the disambiguation process.

The first table (WordTbl) holds frequencies of all morphological parses of words, and the second one (SuffixTbl) holds the frequencies of all possible morphological parses for suffixes. Of course, WordTbl also indicates most likely morphological parses of words since the highest frequency morphological parse is the most likely parse. Since all possible Turkish words cannot be seen in a training corpus, WordTbl will not hold most likely parses for all words. In order to make

<sup>1</sup> Noun is a major word category; A3sg is a noun agreement marker; Pnon is a noun possessive marker; Nom is a noun case marker; derivational boundaries are marked with  $\wedge$ DB.

an intelligent guess for the most likely parse of an unseen word, we use its suffix. For this purpose, we create SuffixTbl. In order to create SuffixTbl, we find suffixes of words according to their correct morphological parse and calculate the frequencies for tags corresponding to suffixes. For example, the suffix of the word “çiçekçi” (florist) whose morphological parse is given in (1) is “çi” and its corresponding whole tag is given in (2). We find frequencies of all corresponding whole tags for suffixes to store them in SuffixTbl.

### 3.1 Learning disambiguation rules

In order to learn disambiguation rules, we use a variation of Brill tagger. After all words in the training corpus are initially tagged with their most likely parses using WordTbl, disambiguation rules are learned. The learned disambiguation rules are based on morphological parses, whole tags, or final tags. The general format of a disambiguation rule is as follows:

*if* conditions *then*  
*select* MPs containing TAG for  $word_i$

The conditions of a rule depend on the possible MPs of the target word  $word_i$  and the current selected MPs of the previous (or following) one or two words. Thus, the conditions of a rule can be one of the following:

- $wordC_i$  **and**  $wordC_{i-1}$
- $wordC_i$  **and**  $wordC_{i-1}$  **and**  $wordC_{i-2}$
- $wordC_i$  **and**  $wordC_{i+1}$
- $wordC_i$  **and**  $wordC_{i+1}$  **and**  $wordC_{i+2}$

Each condition  $wordC_k$  is in the following form:

TAG of  $word_k = TAG_a$

TAGs appearing in the conditions or the MP selection part of a rule can be MPs, whole tags, or final tags.

In the learning of disambiguation rules, a variation of Brill tagger (Brill, 1995) is used. All possible rules are tried in order to select the rule that gives the best improvement. After applying the selected rule, we repeat the process in order to infer the other rules. These iterations end if there is no progress or the improvement is below a threshold. In the selection of the best rule, our method differs from the original Brill tagger. We select the rule with the highest precision as the best rule in iterations. For example, if rule A causes 100 correct tags and 1 wrong tag and rule B causes only 10 correct tags without any wrong tags, the original Brill tagger may choose the rule A for that iteration. However, we select rule B

because it causes no mistakes. The reason for this approach is that we want to increase the correctness of the condition words in the rule applications for later steps of the algorithm, and mistakes in early stages can cause more mistakes in further steps.

The rules are learned using the dataset of 25098 hand-tagged words. After tagging all words in the training set with their most likely tags, we infer the best rule at each iteration step of the algorithm. We generate all possible rules from all the words in the dataset. After generating all rules, we select the rule with the highest precision as the best rule. If there is more than one rule with the highest precision, we select the one, which affects more words. When there is more than one rule with the highest precision and they affect the same number of words, we select any one of them. We applied ten-fold cross-validation for experiments. In the training part of the experiments, we have learned 395.2 rules on average.

### 3.2 Morphological Disambiguation

In the morphological disambiguation of Turkish words, we have used a hybrid disambiguation system, which uses statistical techniques, rule based techniques and some heuristics. After the given Turkish text is morphologically analyzed by a Turkish morphological analyzer, the hybrid disambiguation steps are applied.

In the morphological analysis of a given Turkish text, SupervisedTagger software (Daybelge and Cicekli, 2007) which uses a PC-Kimmo based morphological analyzer (Istek and Cicekli, 2007) is used. SupervisedTagger contains a morphological analyzer and a rule-based morphological disambiguation tool. The morphological parsing capability of SupervisedTagger is improved by using an updated unknown word recognizer and new heuristics for proper nouns and foreign words. If a word begins with a capital letter and it is not the first word of the sentence, it is assumed that it has also a proper name morphological parse even though it is not in the proper name list. In addition, the words that are not correct according to the Turkish grammatical rules are assumed to be foreign words that have proper name morphological parse. By these extensions, the average number of morphological parse per word is increased from 1.8 to 2.0.

In our hybrid disambiguation tool, we use the statistical information in tables WordTbl and SuffixTbl, hand-crafted rules of SupervisedTagger, rules learned by our Brill tagger and

some fall-back heuristics. The disambiguation algorithm consists of five major components:

- *Selection of the Most Likely Tag of Word*
- *SupervisedTagger Disambiguation*
- *Selection of the Most Likely Tag of Suffix*
- *Application of the Learned Rules*
- *Selection with Fall-Back Heuristics.*

The system tries to find the correct morphological parses step by step using the components in the given order. Correct parses of words can be selected or ambiguity levels of words can be reduced by eliminating some illegal parses. But words can be still ambiguous after intermediate steps. After the last step *Selection with Fall-Back Heuristics*, a single morphological parse will be definitely selected for each word.

#### ***Selection of the Most Likely Tag of Word (MW)***

- The statistical information in WordTbl helps us to find the most likely parses of words appearing in the training set. If the word exists in WordTbl, the most frequent parse is selected for that word. Since not all words appear in the training set, some words will be still ambiguous at the end of this step. WordTbl may not contain all words because our training data set is small, and the number of unique Turkish words is huge. In one of our experiments, we determined that the number of unique words is 870,000 in a 6 billion word Turkish corpus. In fact, this is one of the reasons that we decided to use a hybrid approach for the morphological disambiguation.

***SupervisedTagger Disambiguation (ST)*** – In this step, the words are tried to be disambiguated by SupervisedTagger software. Supervised-Tagger uses 342 hand-coded disambiguation rules of two types: *selection* and *elimination* rules. The selection rules select a morphological parse directly. The elimination rules eliminate the wrong ones as much as it can. In other words the selection rules completely disambiguate words, and the elimination rules reduce the ambiguity levels of words. SupervisedTagger is applied only to ambiguous words. At the end of this step, there can still be ambiguous words but the ambiguity level can be reduced by the rules of SupervisedTagger.

#### ***Selection of the Most Likely Tag of Suffix (MS)***

– If the word is not disambiguated by the first two steps, we try to disambiguate using the statistical information in SuffixTbl. The possible suffixes of a word are determined according to its morphological parses, and the most likely morphological parse corresponding to those suf-

fixes is selected if the suffixes appear in SuffixTbl. The word may not be disambiguated at this step because of the huge number of possible suffixes. In one of our experiments, we also determined that the number of unique suffixes is 40,000 in a 6 billion word Turkish corpus.

***Application of the Learned Rules (LR)*** – It can be considered that tagging words according to their frequency is not correct. In this step, we are trying to correct our mistakes and handle special cases by applying the rules that are learned by our Brill tagger. The order of rule application is the order of learning. The condition part of a rule contains a condition depending on the target word of the rule, and one or two more conditions depending on other condition words. A rule can be applicable to a target word if all of the following conditions are satisfied:

- Its condition words are completely disambiguated and satisfy their conditions.
- The target word is disambiguated and satisfies its condition, or the target word is ambiguous and one of its still possible parses satisfies its condition.
- At least one of the parses of the target word contains the correct tag given in the selection part of the rule.

When a rule is applied, the target word can be completely disambiguated, or some of its parses are selected as its possible parses. If the target word contains only one morphological parse satisfying the correct tag, it is disambiguated; otherwise its parses satisfying the correct tag are selected as possible parses for the next step and others are eliminated. For example, the following rule is applicable under the given conditions:

***if*** (final TAG of word<sub>i</sub> = *Adverb* **and**  
whole TAG of word<sub>i-1</sub> = *Noun+A3sg+P3sg+Nom*)  
***then select*** MPs with whole tag *Adjective* for word<sub>i</sub>

If the whole tag of the selected MP of word<sub>i-1</sub> is *Noun+A3sg+P3sg+Nom*, the final tag of at least one of possible MPs for word<sub>i</sub> is *Adverb*, and word<sub>i</sub> contains at least one MP having the whole tag *Adjective*, then MPs containing *Adjective* tag are selected for word<sub>i</sub>.

***Selection with Fall-Back Heuristics (SH)*** - At this last step, a small number of words can still be ambiguous. In this step, we perform the selection with fall-back heuristics in order to disambiguate the remaining ambiguous words. We have determined the following four heuristics and applied them in the given order. The application order is determined empirically.

a) *Selection of Non-Derived (SND)* – SND heuristic selects the parses containing no derivation suffixes since non-derived words are more common than derived words.

b) *Selection of Proper Noun (SP)* - SP heuristic selects the proper noun senses of the words if their possible parses contain proper noun senses.

c) *Selection of Noun (SN)* - SN heuristic selects the parses that their part of speech tags are noun.

d) *Selection of Shortest (SS)* - After applying all techniques and heuristics, if the word is still not disambiguated, we select the shortest parse in terms of the character length.

Number of words	25098
Number of distinct words	8493
Average number of parses per word	1.982
Number of words with single parse	12260
Maximum number of parses in one word	16
Number of distinct parses	17934
Number of distinct whole tags	2052
Number of distinct final tags	343
Number of proper nouns	3305
Number of non-proper nouns	9670
Number of derived words	4772

Table 1. Statistics of Data Corpus

## 4 Evaluation

We have constructed a data corpus consisting of 25098 hand-tagged words. In the preparation of the corpus, we used Turkish texts from different news portals. Ten graduate students tagged words with correct morphological parses using SupervisedTagger software. The statistical information about our dataset is given in Table 1. There are 12 different part of speech tags which are noun, proper noun, conjunction, pronoun, adjective, question, interjection, verb, adverb, post-position, number and punctuation. The 48.8% of the corpus is unambiguous. The most ambiguous word has 16 different parses. There are 2052 distinct whole tags, which show the ambiguity problem of Turkish.

Our disambiguation system uses five different techniques (MW, ST, MS, LR, and SH) step by step. It is obvious that the order of the techniques is crucial for the performance of the system. In order to see which order gives the best accuracy, we have applied each technique separately and obtained the average accuracies by using 10 fold cross validation. In Table 2, the second column shows the average number of words having more than one parse and they are processed by the corresponding technique. The accuracy of the

technique for the applied words is given in the third column. The fourth column shows the accuracy for disambiguated words so far (disambiguated words by the technique plus unambiguous words (UW)).

Technique	# of words applied	Acc. of Tech.	Acc. of (UW+Tech.)
MW	822.8	0.916	0.966
ST	802.4	0.798	0.919
MS	872.4	0.700	0.873
LR	26.6	0.744	0.994

Table 2. Results of techniques for the first step

Since MW gives the highest accuracy, it is reasonable to choose MW for the first step. Applying the learned rules at the first step is not reasonable since there are not enough disambiguated words yet. The reason for having high accuracy so far is that we have few words that are disambiguated in the first step, and unambiguous words in the corpus increase the accuracy.

Technique	# of words applied	Acc. of Tech.	Acc. of (UW+MW+Tech)
LR	18.9	0.741	0.967
ST	260.5	0.873	0.955
MS	369.4	0.761	0.934

Table 3. Results of techniques for the second step

For the second step, we tried MS, ST, and LR. The results are given in Table 3. When we compare Table 2 and Table 3, we can see that the accuracy of techniques increased, meaning that using more reliable techniques in the early steps causes an increase in the accuracy of other techniques by eliminating words that they cannot disambiguate correctly. Applying the learned rules (LR) at this step is again the worst technique. Using ST in the second step gives a higher accuracy than using MS. It is better to disambiguate more words in earlier steps with higher accuracy since the ambiguous words will be disambiguated with less reliable heuristics unless we disambiguate them at earlier steps. Thus, we select ST as the second, and MS as the third. Since ST and MS are better than LR, LR is chosen as the 4th component. The accuracy at the end of the 4th step is 0.942.

After the applications of the first four components, there are still some ambiguous words, and the number of ambiguous words after applying MW, ST, MS and LR is 71.4 on average (2.8%).

In order to disambiguate the remaining ambiguous words, we use fall-back heuristics. Since SND-SP-SN-SS order for the fall-back heuristics produced the best accuracy, we use that order. Finally, we disambiguated all words having the accuracy of 0.934 by using the order of MW-ST-MS-LR-SH.

SupervisedTagger uses handcrafted disambiguation rules. In order to measure the performance of the statistical components of our system, SupervisedTagger component is removed. The accuracy of the overall system is dropped from 0.934 to 0.924. This means that handcrafted rules help to improve the performance of the system. We believe that the importance of handcrafted rules will reduce significantly if we train our system with a huge tagged corpus.

In the calculation of the accuracy, we consider the whole morphological parse. However, in some words, all parses have same inflections after their last derivations so that they have the same grammatical function in the sentence. In other words, they have same final tags. In the calculation of the accuracy, if we consider only the final inflections (the final tags), the accuracy of the overall system becomes 0.941. When only the final part of speech tags are considered, the accuracy becomes 0.969.

Selection						
True	Prop	Adj	Adv	Noun	Verb	
Prop	6.6	5.7	1.2	12.3	2.4	
Adj	1.3	4.2	2.0	6.0	1.5	
Adverb	0.2	3.4	1.3	2.4	0.1	
Noun	18.0	9.7	1.4	57.8	4.3	
Verb	0.9	1.5	0.0	2.2	5.1	

Table 4. The distribution of confusions

When we examined the errors of our system, we observed that most of the mistakes are in nouns, proper nouns, adjectives, verbs and adverbs. In Table 4, the distribution of wrong disambiguation is given. In the calculation, the average number of mistakes in every fold is used. We can see that adjectives are mostly confused with nouns. This is reasonable, since every adjective can also be used as noun in Turkish. Adverbs are also mostly confused with adjectives. Nouns are mostly confused with other nouns. This is an expected result since Turkish is an agglutinative language and there can be many different inflections from a stem. Verbs are most confused with verbs with different inflections. In addition, we can say that nouns are the POS tags mostly confused while adverbs are the least.

In our version of Brill Tagger we prefer the rules with minimum number of errors first instead of preferring the rules with most accuracy increase which the original Brill tagger uses. In order to see whether learning transformation based rules that cause no errors is useful or not, we defined a base method to apply our data set. In this base method, we select the most likely MP for a word from WordTbl if the word is in WordTbl. If it is not in WordTbl, the first MP for the word is selected. That is to say, the most likely MP for a word is selected randomly if it does not appear in the training corpus. Then we applied our learned rules and rules learned according to original Brill Tagger separately in order to see the difference between them. We again applied ten-fold cross validation and our variation had much higher accuracy than the original Brill Tagger.

## 5 Conclusion and Future Work

In this paper, we propose a hybrid disambiguation method that combines the statistical approaches with rule based approaches for Turkish. The first step is the selection of the most likely tags of words. If word is not disambiguated yet, we use hand-crafted rules. Then we use the most likely tags of suffixes for disambiguation. The learned transformation rules are applied in the fourth step.

For training and testing, we have constructed a relatively small corpus, which consists of highly ambiguous words. We applied ten-fold cross validation and obtained 93.4% accuracy on average when we considered whole morphological parses of words. The accuracy increases to 94.1% when final tags are considered. In addition, our accuracy is 96.9% for POS tags. When we use a huge corpus, we believe that our results will improve further.

We have used our components in different orders to see their effects. We observed that MW performs best. MS is better than ST for handling harder words. Considering ST together with the statistical approaches increases the performance of the system. The learned rules have also increased the accuracy.

Our system gives promising results in the disambiguation of Turkish words. Enlarging the corpus which will be useful for the statistical parts is left as a future work. In addition, examining different rule types and learning methodologies are also left for future work.



## References

- E. Brill, Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, *Computational Linguistics*, 21(4):543–565, (1995).
- K.W. Church, A stochastic parts program and noun phrase parser for unrestricted text, *Proceedings of the Second Conference on Applied Natural Language Processing*, pp:136-143, (1988).
- D. Cutting, J. Kupiec, J. Pedersen and P. Sibun, A practical part-of-speech tagger, *Proceedings of the Third Conference on Applied Natural Language Processing*, pp:133-140, (1992).
- W. Daelemans, J. Zavrel, P. Berck and S. Gillis, MBT: A memory-based part of speech tagger-generator, *Proceedings of the Fourth Workshop on Very Large Corpora*, pp:14-27, (1996).
- T. Daybelge, and I. Cicekli, A Rule-Based Morphological Disambiguator for Turkish, *Proceedings of Recent Advances in Natural Language Processing*, pp:145-149, (2007).
- N. Ezeiza, I. Alegria, J.M. Arriola, R. Urizar and I. Aduriz, Combining Stochastic and Rule based Methods for Disambiguation in Agglutinative Languages, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp:379-384, (1998).
- J. Hajic and B. Hladká, Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp:483-490, (1998).
- J. Hajic, Morphological Tagging: Data vs. Dictionaries, *Proceedings of the Applied Natural Language Processing and the North American Chapter of the Association for Computational Linguistics*, (2000).
- D.Z. Hakkani-Tür, K. Oflazer and G. Tür, Statistical Morphological Disambiguation for Agglutinative Languages, *Computers and the Humanities* 36(4), (2002).
- O. Istek and I. Cicekli, A Link Grammar for an Agglutinative Language, *Proceedings of Recent Advances in Natural Language Processing*, pp:285-290, (2007).
- G. Leech, R. Garside and M. Bryan, 1994. CLAWS4: The tagging of the British National Corpus, *Proceedings of COLING*, pp:622-628, (1994).
- M. Levinger, U. Oman and A. Itai, Learning Morpho-Lexical Probabilities from an Un-tagged Corpus with an Application to Hebrew, *Computational Linguistics* 21(3), 383-404, (1995).
- B. Megyesi, Improving Brill's POS Tagger for an Agglutinative Language, *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp:275-284, (1999).
- K. Oflazer and G. Tür, Combining Hand-crafted Rules and Unsupervised Learning in Constraint-based Morphological Disambiguation, *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, (1996).
- K. Oflazer, and G. Tür, Morphological Disambiguation by Voting Constraints, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, (1997).

# A Dynamic Confusion Score for Dependency Arc Labels

Sambhav Jain and Bhasha Agrawal  
Language Technologies Research Center  
IIIT-Hyderabad, India

{sambhav.jain, bhasha.agrawal}@research.iiit.ac.in

## Abstract

In this paper we propose an approach to dynamically compute a *confusion score* for dependency arc labels, in typed dependency parsing framework. This score accompanies the parsed output and aims to administer an informed account of *parse correctness*, detailed down to each edge of the parse. The methodology explores the confusion encountered by the oracle of a data driven parser, in predicting an arc label. We support our hypothesis by empirically illustrating, for 20 languages, that the labels with a high confusion score are notably the predominant parsing errors.

## 1 Introduction

Recently, a major research drive has been towards building data driven dependency parsers for various languages. Shared tasks like CoNLL-X and CoNLL 2007 have acted as development and testing grounds for various efforts in the field. The majority of the emerged systems follow either *graph based paradigm* (McDonald et al., 2005; McDonald and Pereira, 2006) eg. MST Parser (McDonald et al., 2005) or *transition based paradigm* (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004) eg. MaltParser (Nivre et al., 2007).

A time complexity relatively lower than their earlier counterparts makes the above parsers apt for use by real time NLP applications like Machine Translation (Galley and Manning, 2009). However, Popel et al. (2011) in the context of MT pointed out that an incorrect parse can hurt the accuracy of output. Thus, correctness of individual parses becomes a key factor in such a setup.

This calls for measures which can indicate upfront the quality of each individual output. A relevant work in this direction is by Mejer and Crammer (2012). They proposed methods to estimate

*confidence of correctness* of predicted parse in a graph based parsing scenario using MSTParser. Graph-based and transition-based parsers exhibit two very distinctive approaches towards parsing, each having its own strengths and limitations (McDonald and Nivre, 2007; Zhang and Clark, 2008). The diversity in the two techniques motivated us to explore and formulate a similar measure in transition based paradigm. We choose MaltParser<sup>1</sup> (Nivre et al., 2007), which produces a parse tree using a *shift-reduce* based transition algorithm, to work with. It uses *SVM* to train an oracle to predict parsing action. The measures proposed by Mejer and Crammer (2012) can not be straight off applied in transition based parsers, as unlike the graph based approach they commit local operations and thus can not directly produce globally optimum k-best parses.

We propose computing an entropy based label confusion score, dynamically computed and assigned while parsing (the computational details are presented in section 2 of this paper). Since entropy measures the uncertainty in a random variable, we prefer to call the measure, in our approach, *confusion score*. This measure aims to give a more informed picture of the parsed output.

We integrated our approach on top of the current functionality of MaltParser, adjusting it to accredit a confusion score with each arc label predicted in the output. Figure 1 depicts a typical parse from our proposed system where each arc label has been designated a confusion score.

The measure can also be utilized to flag potential *incorrectly parsed edges* which later, can either be manually corrected or altogether discarded (to fall back on lower level but more accurate features). We empirically illustrate in section 4, that such a score can be effectively used for automatic error detection in parsed outputs and guide manual

<sup>1</sup>MaltParser version 1.7 from <http://www.maltparser.org>

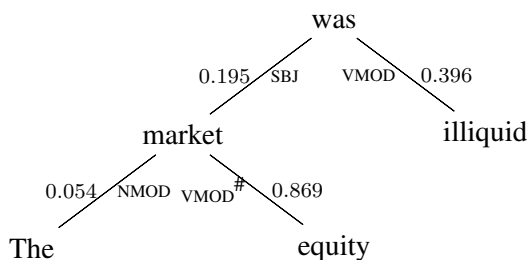


Figure 1: A dependency parse tree with edge confusion score. '#' represents incorrect arc label.

correction.

## 2 Dynamic Confusion Score

MaltParser outputs a single best parse by greedily choosing parsing actions advocated by an oracle trained on the training data. In a typed dependency framework, the parser performs two distinct kinds of actions; attachment of vertices and assigning arc labels to edges. MaltParser provides a choice<sup>2</sup> to train separate oracles for these two kinds or a single oracle that jointly predicts attachment and arc label.

In case of separate oracles for attachment and label prediction, the parser queries the attachment oracle until a *Left Arc* or *Right Arc* action is predicted. The label oracle is then queried for an arc label in the given context. There is further a provision for having a separate oracle for *Left Arc* and *Right Arc* label prediction. Nevertheless, an oracle is always queried for a parsing action against a given context.

### 2.1 Uncertainty in Label Prediction

The problem of predicting arc label correctness can mathematically be posed as follows: For random variables  $X$  and  $Y$  representing context and parsing-actions respectively; an oracle  $\phi$  is defined such that  $\phi : X \rightarrow Y$ . Here,  $Y$  is always a closed set, comprising permissible parser actions. The uncertainty in predicting  $Y$  to one of its possible values  $y_1, \dots, y_n$  can be attributed as the confusion, the oracle has in prediction. It is known that entropy is a measure of the uncertainty in a random variable (Ihara, 1993). Thus, this uncertainty can be quantitatively determined by  $entropy(H)$  calculated by the following formula

$$H(Y/X) = - \sum_{i=1}^n p(y_i/X) \log p(y_i/X) \quad (1)$$

<sup>2</sup><http://www.maltparser.org/userguide.html#predstrate>

were  $p(y_i/X)$  is the posterior probability of  $y_i$  being predicted as the parsing action in the given context  $X$ . The higher the entropy, the more uncertain the oracle is about the prediction.

However, there is no readily available provision indicating the magnitude of confusion the oracle encounters during prediction. The rest of this section presents a sequential account of our approach.

### 2.2 SVM based Oracle

The oracle discussed earlier, is a multiclass classifier which predicts a transition action based on the context. MaltParser employs Support Vector Machine (Cortes and Vapnik, 1995) for classification and provides an option between LIBSVM (Chang and Lin, 2011) and LIBLINEAR (Fan et al., 2008) to build the classifier(s). Both implement “*one-vs-one multi-class classification*” method which incorporates  ${}^nC_2$  binary models ( $n$  denotes number of classes), one for each distinct pair of classes. Prediction is done by voting among these binary classifiers and the class with maximum votes is emitted as decision class. This method does not exert any sort of probabilities and in our scenario we seek posterior probability estimates of the classes.

### 2.3 Posterior Probabilities

Platt et al. (1999) showed that posterior probabilities can be estimated in SVM by training the parameters of an additional sigmoid function to map the SVM output into probabilities. Later, Wu et al. (2004) extended the idea for multiclass probability estimates by combining pairwise class probabilities. In our work, we utilize the second method (proposed in Wu et al. (2004)), which suggests the following optimization formula:-

$$\min_p \sum_{i=1}^k \sum_{j:j \neq i} (r_{ji}p_i - r_{ij}p_j)^2$$

$$\text{subject to } \sum_{i=1}^k p_i = 1, p_i \geq 0, \forall i$$

where,

$k$  = total classes,

$r_{ij}$  = probability of  $i$  in binary classifier with classes  $i$  &  $j$ ,

$p_i$  = probability of  $i$  in multiclass estimation

The solution to above optimization, furnishes multiclass estimation of probability for each class.

## 2.4 Entropy as Confusion Measure

Now, with the available class posterior probabilities, entropy based confusion measure is computed using equation 1. The base of the  $\log$  is the number of label classes which ensures the entropy to be in the range of 0 to 1. Confusion score for arc label would require querying the arc label oracle, and thus MaltParser must be configured to deliver separate oracles for attachment and arc label in the training phase.

## 3 Extendibility of the Confusion Score

This section presents the possible extensions and constraints of the proposed score.

### 3.1 Extension to Full Parse Confusion

The confusion score in the proposed approach is calculated separately for each arc label. Calculation of a confusion score for a full parse can be done by taking an average over the edges in the parse. Other measures like average of worst  $k$  labels, score of worst label itself, etc. can also be adopted. This enables visualization of the confusion for the complete parse tree. This can be apt in the scenario of a large collection of parse trees, such as treebanks and also for applications like Active Learning (Tang et al., 2002; Hwa, 2004).

### 3.2 Extendibility to Other Algorithms

Since our method executes at the oracle level, it is independent of the algorithmic choice used in the parser. Also, pseudo projective transformation (Nivre and Nilsson, 2005) too is an extrinsic process, so non-projectivity does not perturb our approach.

### 3.3 Extendibility for Attachments

The approach, at first, may seem extendable to attachments also since it would require querying the attachment oracle for attachment confusion. However it is not extendable to edge correctness prediction. The restricting factor being the presence of non-labeling parser action i.e. *Shift* and *Reduce*. Since these transitions are not decomposed over the tree edges, the oracle confusion associated with them can not be delineated to any edges. For example, at a given point in the parsing process, assuming the arc-standard system (but the same holds for other algorithms also), the oracle will need to decide if it should perform a *Left*,

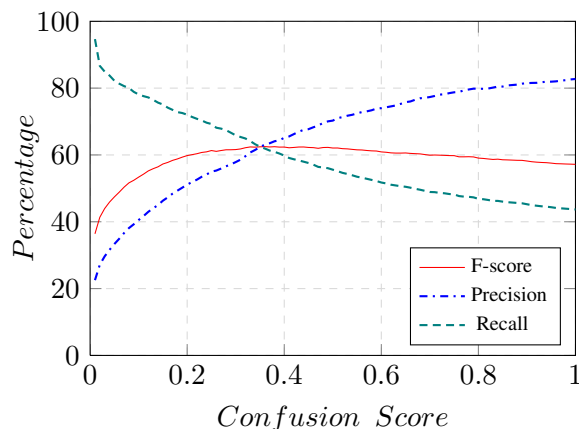


Figure 2: Precision, recall and  $f$ -score for various values of confusion score on ‘Hungarian’ development set.

*Right* or *Shift* action. This decision will influence not only the single edges added by the left or right operation, but also other, future edges. It should be noticed that the Shift action will not add any edge, but will have a very complex effect on the set of edges that could or could not be added in the future. Thus, the entropy of this particular decision cannot be attached to any specific final edge and hence the methodology can not be extended to arc-attachments.

## 4 Error Detection in Parser Output

In this section, we empirically illustrate the efficacy of our proposed measure in automatic error detection and guiding manual error correction.

### 4.1 Automatic Error Detection

Automatic error detection aims to efficiently determine and flag incorrectly predicted edges. The edges exhibiting a high confusion score are also highly probable to be incorrect, as the oracle is uncertain in its decision. Using this insight, an edge-label is flagged as potential error if its confusion score is above a pre-calculated threshold( $\theta$ ).

In this task we have focused on the arc label correctness, i.e. we flag the edges which have a high probability for an incorrect arc label.

### 4.2 Data and Experimental Setup

We conducted experiments on 20 languages, using data from CoNLL-X (Buchholz and Marsi, 2006), CoNLL 2007 (Nilsson et al., 2007) and MTPIL COLING 2012 (Sharma et al., 2012) shared tasks on dependency parsing. We carried out experiments on the systems proposed in Nivre et al.

Language	Threshold ( $\theta$ )	F-score (%)	Precision (%)	Recall (%)	EDI		
					1% edges(%)	5% edges(%)	10% edges(%)
Arabic <sup>#</sup>	0.14	50.71	41.39	65.45	4.43	20.98	36.68
Basque <sup>#</sup>	0.16	51.74	46.62	58.13	2.57	9.58	24.41
Catalan <sup>#</sup>	0.11	48.67	48.54	48.79	7.63	26.53	44.79
Chinese <sup>#</sup>	0.09	41.35	41.68	41.04	5.70	20.02	36.00
Czech <sup>#</sup>	0.08	51.61	47.85	56.02	4.20	18.82	35.92
English <sup>#</sup>	0.09	47.71	57.78	40.63	8.54	33.89	44.58
Greek <sup>#</sup>	0.12	54.47	44.76	69.54	4.84	20.28	35.89
Hungarian <sup>#</sup>	0.36	61.80	69.64	55.54	6.23	31.14	53.49
Italian <sup>#</sup>	0.15	43.48	39.43	48.45	2.57	19.73	40.37
Turkish <sup>#</sup>	0.14	53.58	43.38	70.05	3.99	21.90	40.82
Hindi <sup>♠</sup>	0.16	49.80	43.57	58.11	4.86	20.48	35.09
Bulgarian <sup>¶</sup>	0.12	47.52	40.45	57.60	7.85	34.88	55.63
Danish <sup>¶</sup>	0.12	49.77	41.85	61.41	6.26	30.39	50.32
Dutch <sup>¶</sup>	0.11	50.29	47.46	53.47	2.63	15.39	34.25
German <sup>¶</sup>	0.09	44.44	37.33	54.91	4.37	23.62	45.64
Japanese <sup>¶</sup>	0.09	41.43	29.12	71.75	4.90	24.49	<b>57.59</b>
Portuguese <sup>¶</sup>	0.11	48.13	44.61	52.25	<b>10.43</b>	<b>35.23</b>	54.51
Slovene <sup>¶</sup>	0.14	54.29	44.84	68.78	5.72	19.38	34.15
Spanish <sup>¶</sup>	0.09	40.00	31.10	56.03	7.43	29.21	46.33
Swedish <sup>¶</sup>	0.13	48.47	42.36	56.63	5.03	24.04	42.37
<b>Average</b>	-	<b>48.96</b>	<b>44.19</b>	<b>57.23</b>	<b>5.51</b>	<b>24.00</b>	<b>42.44</b>

Table 1: Language wise results for automatic error detection task. EDI x% edges= Error detected on inspecting top x% of total edges. Data Source:- ¶:CoNLL-X, #:CoNLL 2007, ♠:MTPIL COLING 2012

(2006), Hall et al. (2007) and Singla et al. (2012), which are individually, the best performing Malt-Parser based systems, in the respective shared tasks. All the results reported here are on the official test sets.

### 4.3 Identifying Optimum Threshold( $\theta$ )

Threshold( $\theta$ ) is a crucial parameter in the experimental setup. An optimum  $\theta$  is chosen by making use of the development set. We iteratively increase candidate values for  $\theta$ , from minimum to maximum possible value of confusion score, with an adequate interval. Corresponding to each of these values, the incorrect edges are flagged and *precision*, *recall* & *F-score* (Manning et al., 2008) are calculated. The value asserting the maximum *F-score* is chosen as the final  $\theta$ . Here for simplicity, we have used balanced *F-score*, i.e.  $F_1$ -score. However, as per the application and available resources, a relevant  $F_\beta$  can be chosen to maximize the yield on the input effort.

$$F_\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}$$

Figure 2 depicts *precision*, *recall* and  $F_1$ -score corresponding to each candidate value

of  $\theta$  for Hungarian development data. The maximum  $F_1$ -score is attained at 0.36 which is thus taken as the final  $\theta$  for Hungarian.

Since, CoNLL-X and CoNLL 2007 datasets do not provide development sets, we hold out random 10% sentences of a training set as development data. The remaining training data is utilized to train a parser and identify an optimum threshold on the development set, as explained earlier. However, the final training is performed on the entire training data and evaluated on the test set.

### 4.4 Results and Discussion

Table 1 exhibits the results obtained for automatic error identification. An average  $F_1$ -score of 48.96%, *precision* of 44.19% and *recall* of 57.23% is obtained over 20 languages in the task.

To efficiently capture the efficacy of our approach, another metric is presented (columns 6-8) which corresponds to the percentage of errors detected by inspecting top 1%, 5% and 10% of total edges. The metric gives a more precise picture of the effort required to correct errors.

Our experiments indicate that on average 42.44% errors can be detected by just inspecting top 10% of total edges. This portrays that the

effort required here is one fourth as compared to that in conventional sequential correction. On inspecting top 5% and 1% of all edges, 24.00% and 5.51% errors can be detected. Best results are obtained for Japanese and Portuguese where 57.59% of errors are detected by merely inspecting 10% of total edges for Japanese, while 35.23% and 10.43% errors are detected on inspecting 5% and 1% edges respectively for Portuguese.

A comparison with Mejer and Crammer (2012) is not possible as they only give confidence scores for parent-child attachments while our approach gives confusion scores for parent-child edge's dependency label. In a typed dependency framework both attachments and labels are significant and hence our approach is complementing Mejer and Crammer (2012).

## 5 Conclusion and Future Work

This paper presents our effort towards computing a confusion score that can estimate, upfront, the correctness of the dependency parsed tree. The confusion score, accredited with each edge of the output, is targeted to give an informed picture of the parsed tree quality. We supported our hypothesis by experimentally illustrating that the edges with a higher confusion score are the predominant parsing errors.

Not only parsed output, manual treebank validation too can benefit from such a score. An n-fold cross validation scheme can be adopted, in this case, to compute and assign confusion scores and detect annotation errors. Also, this score has scope in active learning where unannotated instances exhibiting high confusion can be prioritized for manual annotation.

## References

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics.

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.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

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.

Michel Galley and Christopher D Manning. 2009. Quadratic-time dependency parsing for machine translation. 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 773–781. Association for Computational Linguistics.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülşen Eryiit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? a study in multilingual parser optimization. *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 933–939.

Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

Shunsuke Ihara. 1993. *Information theory for continuous system*, volume 2. World Scientific Publishing Company.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Ryan T McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*, pages 122–131.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, volume 6, pages 81–88.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.

Avihai Mejer and Koby Crammer. 2012. Are you sure?: confidence in prediction of dependency tree edges. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 573–576. Association for Computational Linguistics.

Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 915–932. sn.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 99–106. Association for Computational Linguistics.

- Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of english text. In *Proceedings of the 20th international conference on Computational Linguistics*, page 64. Association for Computational Linguistics.
- Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 221–225. Association for Computational Linguistics.
- Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kubler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95.
- John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- Martin Popel, David Mareček, Nathan Green, and Zdeněk Žabokrtský. 2011. Influence of parser choice on dependency-based mt. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 433–439. Association for Computational Linguistics.
- Dipti Misra Sharma, Prashanth Mannem, Joseph van-Genabith, Sobha Lalitha Devi, Radhika Mamidi, and Ranjani Parthasarathi, editors. 2012. *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*. The COLING 2012 Organizing Committee, Mumbai, India, December.
- Karan Singla, Aniruddha Tammewar, Naman Jain, and Sambhav Jain. 2012. Two-stage approach for hindi dependency parsing using maltparser. *Training*, 12041(268,093):22–27.
- Min Tang, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 120–127. Association for Computational Linguistics.
- Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. 2004. Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*, 5:975–1005.
- Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of 8th International Workshop on Parsing Technologies*, pages 195–206.
- Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–571. Association for Computational Linguistics.

# Increasing the quality and quantity of source language data for unsupervised cross-lingual POS tagging

Long Duong<sup>1,2</sup>, Paul Cook<sup>2</sup>, Steven Bird<sup>2</sup> and Pavel Pecina<sup>1</sup>

<sup>1</sup>Faculty of Mathematics and Physics, Charles University in Prague

<sup>2</sup>Department of Computing and Information Systems, University of Melbourne

lduong@student.unimelb.edu.au, paulcook@unimelb.edu.au,

sbird@unimelb.edu.au, pecina@ufal.mff.cuni.cz

## Abstract

Bilingual corpora offer a promising bridge between resource-rich and resource-poor languages, enabling the development of natural language processing systems for the latter. English is often selected as the resource-rich language, but another choice might give better performance. In this paper, we consider the task of unsupervised cross-lingual POS tagging, and construct a model that predicts the best source language for a given target language. In experiments on 9 languages, this model improves on using a single fixed source language. We then show that further improvements can be made by combining information from multiple source languages.

## 1 Introduction

Supervised part-of-speech (POS) taggers perform very well in cases where substantial manually-annotated data is available, as is the case for languages such as English, Portuguese, German, French and Arabic. For example, Petrov et al. (2012) built supervised POS taggers for 22 European languages using the TNT tagger (Brants, 2000), with an average accuracy of 95.2%. However, creating annotated linguistic resources is expensive and time-consuming. Many widely-spoken languages, such as Vietnamese, Javanese, and Lahnda have little or no manually annotated data, making a supervised approach impossible.

However, parallel texts are becoming increasingly available through sources such as multilingual websites and documents, and large archives of translation memory from books, news, etc. Moreover, the number of languages with parallel data is increasing. The era of English dominating one side of parallel texts is shifting to a far wider range of languages. Parallel data can

be exploited to bridge languages, and to transfer annotated information from a highly-resourced *source* language to a lesser-resourced *target* language, to build unsupervised POS taggers (e.g., Das and Petrov, 2011; Duong et al., 2013).

One issue in building such a tagger is choosing the source language. English is commonly used, because parallel data which has English on one side is often most readily available. However, the appropriate source language might depend on the target language. For example, Snyder et al. (2008) found that a better tagger for Slovene could be built by using data from Serbian – a closely related language – than from English. Moreover, if parallel data for a target language with more than one source language is available, it might be possible to exploit this additional information; however, this issue has not been explored to date.

In this paper we build unsupervised POS taggers for 72 language pairs. We identify features based on monolingual and parallel corpora that we use to predict the best source language to build a tagger for a given target language. We show that choosing an appropriate source language can improve the accuracy of a state-of-the-art unsupervised POS tagging methodology, compared to using a single fixed source language. This prediction can be done based on features of the source and target language derived from monolingual corpora – important if parallel data is not available for our target language, and we need to choose which data to collect – although further improvements can be obtained using features based on parallel corpora. We then show that if multiple source languages are available, even better accuracy can be obtained by combining information from them.

## 2 Related work

One approach to build an unsupervised POS tagger is to *project* tag information from a resource-rich source language to a resource-poor target lan-



guage. Das and Petrov (2011) and Duong et al. (2013) both achieve state-of-the-art performance on eight European languages using this cross-lingual approach. The two approaches are similar in the following respects. First, both project tag information from source to target language, applying some kind of noise reduction along the way: Das and Petrov use high confidence alignments, while Duong et al. use high confidence sentences. Second, both use a semi-supervised method to obtain more labeled data: Das and Petrov use graph based label propagation, while Duong et al. use self-training. Finally, both apply noise reduction/filtering on the (automatically) labeled data: Das and Petrov only extract the tag dictionary from labeled data, while Duong et al. heuristically revise tags after each self-training step. Crucially, in both of these approaches, once a tagger is built from parallel data, it can be used to tag monolingual text. The method of Duong et al. is less computationally intensive than that of Das and Petrov, as the graph-based propagation algorithm used by the latter requires convex optimisation. Because of its relative simplicity, yet comparable accuracy, in this paper we extend the method of Duong et al.

Both Das and Petrov and Duong et al. exploit the Europarl Corpus with English as the source language (Koehn, 2005).<sup>1</sup> However, as recent work has shown, it is worth considering other choices of source language. For example, Snyder et al. (2008) found that the accuracy of a Slovene tagger improved by 7.7% when paired with Serbian, a closely related language, but only 1.3 percentage points when paired with English. Reddy and Sharoff (2011) and Hana et al. (2004) showed that for closely related languages, transition probabilities for an HMM tagger can be used interchangeably. This suggests that the source language might have a drastic effect on tagger performance. In this paper we investigate the problem of making a good choice of source language(s).

### 3 Parallel data

We would like to conduct experiments on a resource-poor target language, however, it would be much harder to evaluate. We instead experiment with nine languages: English, Danish, Dutch, Portuguese, Swedish, Greek, Italian, Spanish, and German. We use the JRC-Acquis corpus which provides parallel data for every pair of 22

<sup>1</sup>Das and Petrov also use the ODS United Nations dataset.

Language	No. of Texts	No. of Words ( $\times 10^6$ )
en	23545	55.5
da	23624	50.9
nl	23564	56.8
pt	23505	59.6
sv	20243	47.0
el	23184	55.9
it	23472	57.2
es	23573	62.1
de	23541	50.9

Table 1: The number of texts and words for each language considered in the JRC-Acquis corpus.

Language	Corpus Size		Voc. Size
	JRC-Acquis	Europarl	
en	-	-	14810
da	1000785	1968800	29867
nl	1132352	1997775	21316
pt	1121460	1960407	19333
sv	1061156	1862234	29403
el	792732	1235976	34992
it	1122016	1909115	19310
es	1117322	1965734	18496
de	1136452	1920209	29860

Table 2: Corpus size (number of tokens) for each language, with English as the source language. The vocabulary size for a 1M word sample from JRC-Acquis for each language is also shown.

European languages (Steinberger et al., 2006). We thus, extract a subset of 72 language pairs. It’s worth nothing that we consider  $(x-y)$  and  $(y-x)$  to be distinct language pairs. To the best of our knowledge, JRC-Acquis is the biggest corpus providing parallel data for all language pairs we consider. Table 1 shows some statistics about the data.

## 4 Features

In this section, we consider factors that influence the choice of source language. We divide the features into two categories: *monolingual features* which exploit only monolingual data, and *bilingual features* which exploit parallel data.

### 4.1 Monolingual features

**Morphological complexity.** Morphologically rich languages introduce complexity when aligning parallel data because there is much greater ambiguity in alignment. Given the reliance of our approach on alignments, morphological complexity is an important factor to consider. We can estimate morphological complexity by counting the number of types, i.e. the vocabulary size, in a fixed amount of text. Table 2 shows the vocabulary size for each language, based on a one

million word sample from JRC-Acquis (although any monolingual corpus could be used).

**Language relatedness.** Our nine languages belong to three language families: Germanic (English, Danish, Dutch, Swedish, German); Romance (Portuguese, Italian, Spanish), and Baltic (Greek). Duong et al. (2013) note that their tagger performs better on Germanic languages than that of Das and Petrov (2011), which might be because this is the same family as the source language used (English). Thus, language relatedness is an important factor to consider.

We quantify language relatedness using lexicostatistics on the Swadesh 200 Wordlist (Dyen et al., 1992). Lexicostatistics involves the judgment of a linguist about whether a given pair of words are cognates. The relatedness of two languages is just the percentage of cognates in the wordlist. Dyen et al. provides a table showing this number for all 84 Indo-European languages. We thus, extract a subset of 36 language pairs from this list.<sup>2</sup> Note that this measure is symmetric.

## 4.2 Bilingual features

**Corpus size.** The most obvious factor is corpus size. The more data we have, the better. We count the number of parallel sentences in the corpus. Table 2 shows the corpus size for each language pair with English as the source side.

**One-to-One alignment proportion.** We believe that one-to-one mappings are more meaningful for this task than many-to-one mappings. The intuition is that, if there is only one possible way to copy a tag from the source language to the target language, we can be more confident about the mapping. The proportion of 1–1 mappings is calculated using a fixed number of parallel sentences (800k sentences) for all language pairs.

**Sentence alignment score.** Sentence alignment scores are provided by the aligner for IBM Model 3. Duong et al. (2013) used these scores to rank sentences in building their tagger, showing this to be effective in choosing high quality sentences. Higher alignment scores might therefore correspond to a more accurate tagger. We use the average sentence alignment score for each language pair as a feature.

**Lexical translation entropy.** We adopt the idea of translation model entropy from Koehn et al.

<sup>2</sup>This estimate of language relatedness is not based on parallel text, and is therefore considered a monolingual feature.

(2009). However, instead of scanning all possible sentence segmentations and calculating the phrase-based entropy, we use a simpler method based on the lexical translation table. That is, the entropy for each lexical entry is calculated as

$$H(s) = - \sum_{t \in T} p(t|s) \times \log_2 p(t|s)$$

where  $T$  is the set of possible translations of word  $s$ , and  $t$  is a translation. For each language, we pick a fixed amount of text (1 million words) and calculate the average entropy for all words.

## 5 Build taggers

In this section we construct 72 taggers, using parallel data for 72 language pairs, and then evaluate the performance of each pair. We use an open source unsupervised cross-lingual POS tagger (UMPOS) from Duong et al. (2013), a state-of-the-art system. UMPOS employs the consensus 12 Universal Tagset (Petrov et al., 2012),<sup>3</sup> to avoid the problem of transliterating between different tagsets for different languages, and to enable comparison across languages.

The input for UMPOS is a tagger for the source language,  $Tagger(s)$ , along with parallel data ( $s-t$ ). The source language  $s$  is tagged using  $Tagger(s)$ , and then the tagged labels are projected to the target language  $t$ . Sentences are then ranked, and a seed model tagger  $T_0$  is built on just the high scoring sentences. By applying self-training with revision, a series of new models  $T_1, T_2, \dots, T_m$  is constructed where  $T_i$  is the tagger after  $i$  iterations. The target language tagger,  $Tagger(t)$ , is then the last model,  $T_m$ .

$Tagger(s)$  is trained from manually annotated data  $Data(s)$  which is mainly derived from the CoNLL 2006 and CoNLL 2007 Shared Tasks. Using the matching provided by Petrov et al., we map the individual tagsets to the Universal Tagset. We train a supervised POS tagger  $Tagger(s)$  on the annotated data using the TNT tagger (Brants, 2000). Table 3 shows the source and size of annotated data, and the 5 fold cross-validation accuracy of  $Tagger(s)$ , for each language.

We evaluate each  $Tagger(t)$  using  $Data(t)$ ; results are shown in Table 4. The average tagger per-

<sup>3</sup>NOUN, VERB, ADJ, ADV, PRON (pronouns), DET (determiners and articles), ADP (prepositions and postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), “.” (punctuation), and X (all other categories, e.g., foreign words, abbreviations).

		Target language									Average
		en	da	nl	pt	sv	el	it	es	de	
Source language	en	-	76.17	72.97	79.57	<b>73.83</b>	50.38	72.20	75.37	73.95	71.81
	da	55.73	-	53.28	50.53	66.08	34.13	46.03	50.34	53.90	51.25
	nl	<b>75.70</b>	<b>76.31</b>	-	78.92	70.24	54.22	70.49	76.90	<b>79.47</b>	72.78
	pt	72.40	69.49	63.07	-	66.67	61.82	<b>74.23</b>	80.50	64.70	69.11
	sv	66.56	75.82	61.20	65.51	-	52.74	58.93	63.88	64.48	63.64
	el	47.67	49.50	49.75	57.11	46.64	-	47.33	62.29	55.16	51.93
	it	74.50	71.60	68.19	<b>84.50</b>	67.92	47.33	-	<b>81.80</b>	68.28	70.52
	es	68.76	68.83	66.34	80.72	68.83	<b>62.29</b>	74.07	-	70.36	70.03
	de	72.24	74.48	<b>76.54</b>	70.87	66.56	55.16	56.98	70.84	-	67.96
	Baseline	30.28	23.27	24.28	24.53	26.35	24.00	25.09	21.98	26.50	25.14

Table 4: Percentage accuracy for the tagger for each source–target language pair. The best tagger for each target language is shown in bold.

Language	Source	No. of Words	% accuracy
en	WSJ/PennTB	1289k	96.74
da	DDT/CoNLL06	94k	96.20
nl	Alpino/CoNLL06	203k	96.42
pt	Floresta/CoNLL06	206k	96.38
sv	Talbanken/CoNLL06	191k	93.95
el	GDT/CoNLL07	65k	97.68
it	ISST/CoNLL07	76k	94.48
es	Cast3LB/CoNLL06	89k	95.36
de	Tiger/CoNLL06	712k	97.79

Table 3: Source and size of annotated data for each language. The accuracy of each source language tagger is also shown.

formance for each source language is also given. It turns out that choosing Dutch instead of English as the source language gives the best average accuracy. The tagger performance on each target language is much better than the baseline that always picks the most frequent tag for each word.

The Greek tagger performs poorly. From Table 2, Greek is the most morphologically complex language in this set, and has the smallest corpus size, two factors which partially explain why tagger performance for Greek is low whether Greek occupies either the source or target language role.

From Table 4, it seems that taggers perform better if the source and target language are in the same language family. For example, the top four source languages for Danish are Dutch, English, Swedish, and German, and the top two source languages for Portuguese are Italian and Spanish. This confirms the intuition in adding language relatedness features in section 4.

Duong et al. (2013) used English as the source language to build taggers for the same eight other languages. The only difference between these two experiments is that Duong et al. used Europarl (Koehn, 2005) data instead of JRC-Acquis. Table 2 also compares the size of parallel data with

Language	JRC-Acquis	Europarl
da	76.2	85.6
nl	73.0	84.0
pt	79.6	86.3
sv	73.8	81.0
el	50.4	80.0
it	72.2	81.4
es	75.4	83.3
de	74.0	85.4
Average	71.8	83.4

Table 5: Accuracy on JRC-Acquis and Europarl using English as the source language.

English as the source language for JRC-Acquis and Europarl. Given that Europarl is larger, higher performance is expected. Table 5 compares the tagger accuracy for each target language using English as the source language, for the two datasets. As expected, the accuracies are higher for Europarl. However, there is a strong correlation between the results for the two experiments (Pearson’s  $r = 0.7$ ). This suggests that, if we had as much data as Europarl for every language pair (not just English), we would expect all numbers in Table 4 to improve substantially (not only the first row where English is the source language).

## 6 Source language selection

In this section, using features defined in section 4 and tagger performance in Table 4, we build a model that can predict the performance of the target language tagger given a source language.

### 6.1 Individual feature correlation

Table 6 shows the Pearson’s correlation ( $r$ ) and coefficient of determination ( $r^2$ ) of each feature with tagger accuracy.

Surprisingly, the one-to-one alignment proportion is very strongly correlated with tagger performance ( $r = 0.745$ ). Lexical translation entropy

Features	$r$	$r^2$
Source vocabulary size	-0.613	0.376
Target vocabulary size	-0.202	0.041
Language relatedness	0.497	0.247
Corpus size	0.620	0.385
One-to-one alignment proportion	0.745	0.556
Sentence alignment score	0.492	0.242
Lexical translation entropy	-0.590	0.348

Table 6: Pearson’s  $r$  and  $r^2$  for each feature.

has a negative correlation, as expected, because lower entropy leads to a better alignment and therefore better tagger performance. The source language vocabulary size is highly negatively correlated, but that strong relationship is not found for the target language. This suggests that the model is not affected much by the target language, but prefers a morphologically simple source language.

Corpus size also has a high positive correlation, confirming the intuition that more data is better. This strong relationship, together with the negative correlation for morphological complexity, consolidates the explanation above about the poor performance of the tagger for Greek, where the availability of data is very limited, and where Greek has the richest morphology of any language considered.

## 6.2 Building a predictive model

In this experiment we build a model to predict the performance of a target language tagger given a source language. We fit all features into a multiple linear regression model. The  $r^2$  value improved greatly to 0.74, compared to 0.556 for one-to-one alignment proportion, the best individual feature.

We evaluate our model in a leave-one-out cross validation experiment. To build a predictive model for language  $t$ , we remove data in Table 4 associated with  $t$  and train the multiple linear regression model  $model(t)$  on the remaining data. So, given source language  $s$  and  $(s-t)$  parallel data,  $model(t)$  outputs the predicted performance of the tagger trained on  $(s-t)$  parallel data. The correlation of the predicted value with the original value (Table 4) is very high ( $r = 0.81$ ).

We also build another predictive model based solely on monolingual features (morphology complexity and language relatedness). The intuition here is that, if we want to build a tagger for a target language, but only have monolingual data for that language, what parallel data would we want to collect first? This monolingual model also shows a high correlation with the original table ( $r = 0.74$ ). If we only use language relatedness, the correla-

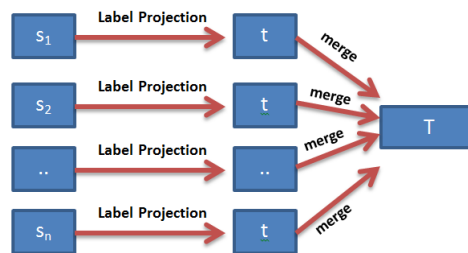


Figure 1: Combining multiple source languages to produce a single file.

tion is very weak ( $r = 0.13$ ), showing that language relatedness on its own is not effective at predicting the best source language.

The predicted best source language for each target language is the language predicted to produce the highest accuracy tagger. Table 7 shows the source language prediction from models exploiting all features, and only monolingual features. The Fixed model always chooses Dutch (nl) as the source language, because Dutch gives the highest average accuracy (Table 4). The Oracle model always picks the best language, and gives the upperbound for the predictive model as a point of comparison. As expected, the model exploiting all features achieves a higher average accuracy than the monolingual model, which nevertheless still outperforms Fixed (although there is some variation for individual languages). With respect to the Oracle upperbound, and Fixed baseline, the error rate reduction for the monolingual and all features models is 10.9% and 52.3%, respectively, showing the effectiveness of using a predictive model.

## 7 Multiple Source Languages

In this section, we combine information from multiple source languages to build a single target language tagger. We take a simple approach to doing so, as shown in Figure 1. Each  $s_i$  is a tagged corpus for source language  $i$ . POS tags are then projected to the target language side  $t$  for each corpus. We merge all of these partially-tagged target language corpora (in which unaligned words are untagged) to form  $T$ .<sup>4</sup> We build the target lan-

<sup>4</sup>Because the JRC-Acquis corpus consists of translations of documents into multiple languages, in some cases the same target language sentence occurs in the parallel corpus for multiple source languages. In this preliminary approach to combining information from multiple source languages, we simply treat these as different target language sentences. Because the sentences are aligned with different source languages, they might contain different partial tag information.

Target language	All features	Monolingual features	Fixed	Oracle
en	pt (72.40)	<b>nl (75.70)</b>	<b>nl (75.70)</b>	nl (75.70)
da	sv (75.82)	en (76.17)	<b>nl (76.31)</b>	nl (76.31)
nl	<b>en (72.97)</b>	<b>en (72.97)</b>	-	de (76.54)
pt	<b>it (84.50)</b>	es (80.72)	nl (78.92)	it (84.50)
sv	<b>en (73.83)</b>	<b>en (73.83)</b>	nl (70.24)	en (73.83)
el	<b>es (62.29)</b>	en (50.38)	nl (54.22)	es (62.29)
it	<b>pt (74.23)</b>	es (74.07)	nl (70.49)	pt (74.23)
es	<b>pt (80.50)</b>	<b>pt (80.50)</b>	nl (76.90)	it (81.80)
de	en (73.95)	en (73.95)	<b>nl (79.47)</b>	nl (79.47)
Average	<b>74.50</b>	73.14	72.78	76.07

Table 7: Best source language prediction (and % accuracy of the corresponding tagger) for models exploiting all features, only monolingual features, and a fixed source language, as well as an oracle model that always picks the best language. The best (non-oracle) source language and accuracy for each target language is shown in bold.

Language	1-best	3-best	5-best	7-best
en	75.70	76.66	76.36	<b>78.16</b>
da	76.31	78.40	<b>82.45</b>	82.43
nl	76.54	76.17	80.00	<b>81.45</b>
pt	84.50	84.91	<b>85.00</b>	84.24
sv	73.83	74.65	74.10	<b>76.66</b>
el	62.29	<b>70.23</b>	67.22	67.69
it	74.23	<b>78.71</b>	78.47	76.05
es	81.80	82.53	82.13	<b>82.64</b>
de	<b>79.47</b>	79.28	77.92	77.35
Average	76.07	77.95	78.18	<b>78.52</b>

Table 8: Tagger accuracy when combining the 1-, 3-, 5-, and 7-best source languages. The best system for each target language is shown in bold.

guage tagger from  $T$  by adapting the method from Section 5. The typical steps for this method are (1) tag the source language, (2) project labels from the source to target language, (3) build the seed model, and (4) apply self-training with revision to produce the final model. Here we simply start from step (3) and build the seed model from  $T$ .

In these experiments we assume that when building a tagger for a target language we have access to all other source languages. Table 8 shows accuracy when combining information from the 1-, 3-, 5-, and 7-best source languages, as determined by an oracle. As more source languages are added, average accuracy increases, demonstrating that the method of Duong et al. (2013) can be substantially improved by combining information from multiple source languages. Having established this, in future work we will consider using the best languages as identified by the various feature sets. Moreover, for individual target languages, the best accuracy is not always achieved using the most source languages, suggesting that further work could be done to identify the best set of source languages. There is also a trade-off be-

tween accuracy and efficiency; taggers built from more source languages are generally slower.

## 8 Conclusions

In this paper, we have investigated the problem of choosing the best source language(s) to use in unsupervised cross-lingual POS tagging based on tag projection in parallel corpora. We have shown that our predictive model can select a source language – based on only monolingual features of the source and target languages – that improves tagger accuracy compared to choosing the single, best-overall source language. However, if parallel data is available, our predictive model is able to leverage this to select a more appropriate source language and obtain further improvements in accuracy. Finally, we showed that if multiple source languages are available, even better accuracy can be obtained by combining information from them.

Based on these findings, a synopsis for building a tagger for a resource-poor target language  $t$  is as follows: (1) if parallel data for  $t$  is unavailable, use monolingual features to predict the best source language  $s$  and collect  $(s-t)$  parallel data; (2) if there are multiple parallel corpora for  $t$ , and there is sufficient time, combine all the corpora to produce a tagger with the best expected accuracy; (3) if time is limited, use all features to identify the  $n$ -best source languages.

In future work, we would like to apply the methods described in this paper for identifying “good” source languages for other cross-lingual NLP tasks which exploit parallel data to transfer annotations between languages, including grammar induction, parsing, and morphological analysis. We further intend to expand our experiments to consider more source and target languages.

## 9 Acknowledgements

This work is funded by Erasmus Mundus European Masters Program in Language and Communication Technologies (EM-LCT) and by the Czech Science Foundation (grant no. P103/12/G084).

## References

- Thorsten Brants. 2000. TnT – A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP '00)*, pages 224–231. Seattle, Washington, USA.
- Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609. Portland, Oregon, USA.
- Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013. Simpler unsupervised POS tagging with bilingual projections. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 634–639. Sofia, Bulgaria.
- Isidore Dyen, Joseph B. Kruskal, and Paul Black. 1992. An Indoeuropean classification: A lexicostatistical experiment. *Transactions of the American Philosophical Society*, 82(5):iii–iv+1–132.
- Jiri Hana, Anna Feldman, and Chris Brew. 2004. A resource-light approach to Russian morphology: Tagging Russian using Czech resources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP '04)*, pages 222–229. Barcelona, Spain.
- 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.
- Philipp Koehn, Alexandra Birch, and Ralf Steinberger. 2009. 462 machine translation systems for Europe. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*. Ottawa, Canada.
- 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. Istanbul, Turkey.
- Siva Reddy and Serge Sharoff. 2011. Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources. In *Proceedings of the Fifth International Workshop on Cross Lingual Information Access (CLIA 2011)*. Chiang Mai, Thailand.
- Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2008. Unsupervised multilingual learning for POS tagging. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 1041–1050. Honolulu, Hawaii.
- Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 2142–2147. Genoa, Italy.

# Towards the Annotation of Penn TreeBank with Information Structure

**Bernd Bohnet**

School of Computer Science  
University of Birmingham  
Birmingham, UK

b.bohnet@cs.bham.ac.uk

**Alicia Burga**

DTIC  
Pompeu Fabra University  
Barcelona, Spain

alicia.burga@upf.edu

**Leo Wanner**

ICREA and DTIC  
Pompeu Fabra University  
Barcelona, Spain

leo.wanner@upf.edu

## Abstract

Information Structure (IS) determines the “communicative” segmentation of the meaning of an utterance, which makes it central to the semantics–syntax–intonation interface and therefore also to NLP. Despite this relevance, IS has not received much attention in the context of the majority of the reference treebanks for data-driven NLP that already contain a semantic and syntactic layers of annotation. We present our work in progress on the annotation of the Penn TreeBank with the thematicity dimension of the IS as defined in the Meaning-Text Theory. We experiment with tagging and transition-based parsing techniques. Especially the latter achieve acceptable accuracy with even very small training samples, which is promising for languages with scarce resources.

## 1 Introduction

The *Information Structure* (IS) (aka *Topic-Focus Articulation*, TFA (Sgall, 1967) in the Prague School and *Communicative Structure*, CommStr (Mel’čuk, 2001) in the Meaning-Text Theory) determines the “communicative” segmentation of the meaning of an utterance. This makes it central to the semantics–syntax–intonation interface (Lambrech, 1994; Hajičová et al., 1998; Steedman, 2000; Mel’čuk, 2001; Erteschik-Shir, 2007) and therefore also to NLP. However, despite its prominence, IS has been largely ignored so far in the context of the reference treebanks for data-driven NLP: Penn Treebank (Marcus et al., 1993) and its semantic counterpart PropBank (Palmer et al., 2005) for English, Tiger (Thielen et al., 1999) for German, Ancora (Taulé et al., 2008) for Spanish, etc. To the best of our knowledge, only the Prague

Dependency Treebank (PDT) (Hajič et al., 2006) is annotated with IS in terms of TFA. This is not to say that no proposals have been made for the annotation of IS in general; see, e.g., (Calhoun et al., 2005) for English, (Dipper et al., 2004) for German, (Paggio, 2006) for Danish, etc. However, in the light of the above mentioned interface, it is crucial to have the same corpus annotated with semantic, syntactic and IS structures.

In this paper, we present our work in progress on the annotation of the ConLL ’09 variant of the Penn TreeBank (PTB) (Hajič et al., 2009) with the *thematicity* dimension of Mel’čuk’s CommStr.<sup>1</sup> We have chosen thematicity because (i) it distinguishes, apart from the traditional Theme and Rheme, a Specifier element, and (ii) it is hierarchical in that a thematicity partition can be embedded into another thematicity partition. Both of these features facilitate a fine-grained communicative partition of complex utterances with subordinations and thus a more accurate and detailed projection between the different layers of the semantics–syntax–intonation interface.

Unlike most other proposals, we aim to automate the annotation procedure and experiment with tagging and transition-based parsing techniques. In this respect, our goal is similar to that of Postolache et al. (2005), who explore different classifier models for automatic labeling of tectogrammatical nodes in the PDT with Topic / Focus. But while Postolache et al. use for training 78.3% (38,737 sentences or 494,759 tectogrammatical nodes) of the TFA+tectogrammatical layer of the PDT, our training sample is infinitely smaller: we train on 360 manually annotated sentences. The purpose of the small training sample is

<sup>1</sup>It is important to note that thematicity does not intend to capture the IS in its entirety—as, e.g., Vallduví (1992)’s or Erteschik-Shir (2007)’s proposals do. It is just one of the eight dimensions of the CommStr, although the most central one.

twofold. First, to minimize the manual annotation effort, and, second, to assess whether automatic IS annotation can be bootstrapped starting from minimal resources.

In the next section, we introduce the theoretical notion of thematicity in the sense of the Meaning-Text Theory and present the criteria for the determination of its individual elements. Section 3 describes our tagging and parsing approaches to automatic annotation of thematicity. Section 4 outlines the experiments we carried out in order to assess the quality of both approaches, and Section 5, finally, discusses the outcome of these experiments and sketches some directions of future work we plan to undertake in this area.

## 2 Theoretical Background

Since its introduction by the Prague School of Linguistics (Mathesius, 1929; Firbas, 1966; Daneš, 1970), a great number of models that define and determine the IS of an utterance have been proposed; for an overview, see, e.g., (Vallduví, 1992; Mel'čuk, 2001; Kruijff-Korbayová and Steedman, 2003; Zimmermann and Féry, 2009). In our work, we draw upon Mel'čuk (2001)'s model, which foresees a tripartite communicative segmentation of the meaning of an utterance: Theme–Rheme–Specifier. The Specifier (SP) sets up the context of the utterance *U*; Rheme (R) denotes the part of *U* that the speaker presents as stated by *U*; and Theme (T) denotes that part of *U* that the speaker presents as something about which R is stated.<sup>2</sup> T and R (also referred to in the literature as *topic/focus* and *topic/comment*) constitute the *Communicative Core* (CC) of a sentence.

The basic unit that we annotate with SP/T/R tags is a *proposition*. A proposition (P) is either a *full clause*, i.e., a clause that contains a finite verb, or a *reduced clause*, i.e., a clause with elision of the corresponding finite verb (as *up 150 in Mitsubishi Estate ended the day at 2680, up 150*). The following assumptions hold: (i) each P possesses a CC; (ii) if a sentence is composed by a coordination of Ps (each with its own subject), no global CC is assigned: each of the Ps has its own CC, with the coordinative conjunction as the SP of the second P; (iii) two non-coordinated Ps (each holding an independent CC, with the subordinated

conjunction as SP) form a global CC such that the one that comes first is T and the other one is R; if their linear order is altered, the T/R assignment switches although the meaning remains the same; (iv) if a sentence contains a main P and a relative subordinate P, one main CC is assigned (without taking into account the presence of the relative clause), and at the same time an embedded CC is assigned to the subordinate proposition. That is, as already mentioned in Section 1, thematicity in the Meaning-Text Theory is *per se* hierarchical in that each thematicity element (SP/T/R) can in itself be again assigned a thematicity structure.<sup>3</sup>

Consider (1) for illustration of the thematicity segmentation of a sample sentence (for clear and unambiguous notation, propositions are enclosed in “{...}”):

(1) {[Years ago]SP, [he]T [collaborated with the new music gurus Peter Serkin and Fred Sherry in the very countercultural chamber group Tashi, {[which]T [won audiences over to dreaded contemporary scores like Messiaen's Quartet for the End of Time]R}.]R }

For the communicative segmentation of a fragment of the PTB as gold standard, we used the following empirically determined criteria:

**Criteria for determining the Specifier:** Given that Specifiers do not express a separate message, but, rather, the context of the message to which they belong, we mark as Specifiers:

- fronted temporal, locative and manner circumstantials: {[Apparently]SP [he]T [did so]R};
- fronted AdjPs with a sentential scope: {[Tired of the same]SP, [he]T [gave up]R};
- fronted discourse markers: {[But]SP [it]T [was neither deep]R};
- circumstantials of the type *according to ...* (independently of their position): {[About 25 % of the insiders]T, [according to SEC figures]SP, [file their reports late]R};
- phrases that introduce direct speech (independently of their position): {[It]T [is done]R, [he said]SP};
- NPs in vocative case (independently of their position): {[Anna]SP, [he]T [did it]R}.

<sup>2</sup>Strictly speaking, Theme and Rheme are defined over the meaning of an utterance, rather than the utterance itself. It is for brevity that we speak of an IS of an utterance.

<sup>3</sup>The hierarchical relations in a given thematicity segmentation are in practice controlled by indices. Thus, ‘T(T1)’ will stand for “theme within the theme element marked as ‘T1’ ” and ‘R(T1)’ for “rheme within T1”. To distinguish between elements of the same type at the same level of the hierarchy, numbers are used: ‘T1’ vs. ‘T2’ ‘R1’ vs. ‘R2’, etc.



**Criteria for determining the Theme:** *Theme* is the part of the sentence that expresses what the Speaker is talking about. Therefore, it tends to be located in the initial part of the sentence (after the Specifiers, if there are any). In an SVO language as English, the Theme therefore coincides most of the times with the subject. Apart from this *ad hoc* criterion, a number of “hard” criteria to identify the Theme are available; among them:

- it can be identified by the question “And what about X” (then, X is Theme): {[John]T [answered the question]R}: *And what about John?* (\**And what about question?*);
- it is not accessible for general negation/questioning: {[He]T [did it]R}: \**Not he did it*;
- a relative clause is treated as an independent proposition, and therefore the relative pronoun is the Theme only if it is subject (otherwise, it is a focalized part of the Rheme): *the boy* {[who]T [cooked]R} vs. *the boy* {[whom]R1-1,Foc [I]T [met]R1-2};
- indefinite pronouns such as *nobody*, *somebody*, *nothing*, etc. and negative noun phrases cannot be Themes: e.g., in *None of the boys did it*, it is not *none of the boys*, which is the Theme, but rather *it*;
- sentences of the form *It + is + Adjective + Infinitive verb* reverse the typical position of the theme, with the infinitive verb being the theme: {[It's necessary]R [to talk]T};
- headings or titles are all-thematic.

Occasionally, a split of Theme can be observed: {[Considered as a whole]T1-1, [[Mr. Lane]T1(SP) [said]R1(SP)]SP, [the filings required under the proposed rules]T1-2 “[will be at least as effective, if not more so , for investors following transactions]R.”}

**Criteria for determining the Rheme:** The easiest way to recognize Rheme is through exclusion: if an element is not Theme nor Specifier, then it is part of the Rheme. A few explicit criteria can also be introduced:

- Rheme can be negated and/or questioned: {[I]T [think so]R}: *I don't think so / Do I think so?*;
- existential sentences (those that begin with *There is/are*) are all rhematic: {[There are apples on the table]R};
- non-fronted temporal, locative and manner circumstantials form part of the Rheme: {[I]T [met John some months ago in the park, in a very unexpected way]R.}

In addition, it is to be noted that if the Rheme

contains a ditransitive verb that allows arguments to exchange syntactic positions, we assume that this exchange is motivated by different ISs; cf. {[John]T [[gave me]T(R1) [money]R(R1)]R1} vs. {[John]T [[gave money]T(R1) [to me]R(R1)]R1} (the tag ‘R’ is supplied with a number for unambiguous notation). Furthermore, within NPs that start with *wh*-words that do not belong to pseudo-clefting constructions, split Rhemes are observed: {[[What]R1-1(T1) [he]T(T1) [said]R1-2(T1)]T1 [was hilarious]R1}.

### 3 Annotating PTB with Thematicity

Given that the thematicity structure as used in this paper is of a hierarchical nature, its automatic annotation can be viewed not only as a tagging but also as a (constituency) parsing task. We carried out experiments with both approaches, taking the tagger variant as baseline (the idea is to apply a simple and well researched technique in order to compare it with a more elaborated one). For both, we use the CoNLL '09 format; cf. Figure 1. A line in this format consists of an id, a word form, a lemma, a pos tag, and the dependency annotation with the head node and the edge label (all retrieved from the CoNLL Shared Task 2009 data set). The last two columns contain the gold communicative tag and the tag predicted in the course of the automatic annotation, respectively.

id	form	lemma	pos	head	edge	com.	p-com.
1	He	he	PRP	2	SBJ	[]T	-
2	believes	believe	VBZ	0	ROOT	[	-
3	in	in	IN	2	ADV	-	-
4	what	what	WP	6	OBJ	[]T	-
5	he	he	PRP	6	SBJ	[	-
6	plays	play	VBZ	3	PMOD	]R]R	-
7	.	.	.	2	P	-	-

Figure 1: Example of a sentence annotated with its thematicity structure in CoNLL format.

In the case of tagging, we aim to assign to each element of a sentence a T, R or a SP tag. For this purpose, we use classifier-based sequence tagging. The tagger assigns one of the three tags to each word by going from left to right through the sentence. For the selection of the appropriate tag, the tagger considers features from a window of two words before and after the word in question.; cf. Table 1 for the features used by the tagger.

For training the classifier of the sequence tagger, we use the perceptron algorithm. Following Collins and Duffy (2002), averaging of the param-

Features based on PoS tags
$\pi(i), \pi(i-1)\pi(i), \pi(i)\pi(i+1), \pi(i+1)\pi(i+2)$
$\pi(i-1)\pi(i-2), \pi(i-2)\pi(i-1)\pi(i),$
$\pi(i-1)\pi(i)\pi(i+1), \pi(i)\pi(i+1)\pi(i+2)$
Features involving PoS tags and word forms
$w(i), \pi(i-1)w(i), w(i)\pi(i+1), w(i+1)\pi(i+2)$
$\pi(i+1)w(i+2), w(i-1)\pi(i-2), \pi(i-1)w(i-2)$
$w(i-1)w(i), w(i)w(i+1)$
$w(i+1)w(i+2), w(i-1)w(i-2)$

Table 1: Features for the sequence tagger.  $i$  ( $i = 0, 1, \dots$ ) denotes the  $i$ th word in the input sentence;  $\pi$  and  $w$  are functors to extract the PoS tags respectively word forms of the tokens.

eters obtained in the training algorithm is applied for classifying the test examples.

In the case of parsing, we aim to derive the hierarchical IS (or *communicative tree*) of a given sentence. The communicative tree  $T_c$  of a sentence  $x = w_1 \dots w_n$  is a quintuple  $T_c = (V, E, L, \delta, 0')$ , such that  $V = V_t \cup V_c$  is a set of nodes, with  $V_t = 0, \dots, n$  as a set of terminal nodes and  $V_c = o', 1', \dots, m'$  as a set of non-terminal communicative (label) nodes;  $E \subseteq V \times V$  is a set of edges;  $L$  is the set of communicative labels (in the case of thematicity: SP, T, R and P);  $\delta : E \rightarrow L$  is a labeling function for nodes;  $0'$  is the root node. That is, we interpret the  $T_c$  as a kind of constituency tree.

For the implementation of the parser, we use the idea of transition-based parsing (Yamada and Matsumoto, 2003; Nivre et al., 2004), which uses a classifier to predict the shift/reduce actions. We draw upon the transition set of the *arc-eager parser* Nivre (2004), but with a slightly different semantics in that we define a transition system for the derivation of the  $T_c$  as a quadruple  $C = (S, Y, c_0, S_y)$ , where  $S$  is a set of parsing states;  $Y$  is a set of transitions, each of which is a (partial) function  $t: S \rightarrow S$ ;  $s_0$  is an initialization function that maps a sentence  $x$  to a configuration  $s \in S$ ; and  $S_y \subseteq S$  is a set of terminal states. A transition sequence for a sentence  $x$  in  $C$  is a sequence of pairs of states and transitions. As set  $S$  of states, we use the tuple  $s = (\Sigma, B, V_c, Z, E, \delta, o)$ , where the stack  $\Sigma$  and the input buffer  $B$  are disjoint sublists of the terminal nodes  $V_t$ ,  $V_c$  is the set of communicative (label) nodes,  $Z$  is the stack of communicative nodes,  $E$  is the set of edges,  $\delta$  is a labeling function for communicative label nodes  $n \in V_c$ , and  $o$  is a counter for the number of pairs of delimitation brackets. The initial state for a sentence  $x$  is  $s_0 = ([0], [1, \dots, n], \{0'\}, [0'], \{\}, \delta, 0)$ . Terminal configurations have an empty buffer and

only the root node  $n$  is contained in the stack  $\Sigma$ :  $s = ([0], [], V_c, Z, E, \delta, x)$ . Figure 2 shows the possible transitions.

As features of the transition-based system, we use a rich feature set based on the dependency structure drawn from (Zhang and Nivre, 2011) (since we use as input a dependency structure these features are available). In addition, we use the path from the top stack element to the word of the last open bracket (as sequence of pos tags). For the training of the transition-based system, we use the perception algorithm with averaging, a beam-search with 10 elements and early update (Collins and Roark, 2004). The oracle for training of the system follows the bottom-up parsing strategy. As soon as the communicative part is completed, we remove (reduce) the nodes that belong to it from the stack. Figure 3 shows a sequence of transitions that the analyser performs to create the  $T_c$  of the example sentence in Figure 1.

## 4 Experiments

Following the criteria in Section 2, four annotators in teams of two manually annotated a fragment of 435 sentences of the PTB with the thematicity structure, in a series of blocks of about 40–50 sentences. To ensure high mutual agreement, the annotation procedure went as follows. First, one of the teams provided a first round annotation of a block of sentences. This annotation was revised by the other team and the two annotations were discussed in plenum to achieve a consensus and to refine the annotation guidelines. The refined guidelines were used by the first team to annotate the next block of sentences—to be again revised by the other team and discussed in plenum. And so on.<sup>4</sup>

For training, we use the first 360 sentences, the next 40 sentences as development set, and the remaining 35 sentences as test set. Table 2 presents the results on the test set for the sequence tagger and the transition-based analyser. The Accuracy Score (AS) measures the correctly assigned thematicity tags on a token basis in the same way as PoS tagging is evaluated. That is, given, e.g., the sequence  $[, -, []T, [, ]R$  and the predicted sequence  $-, -, []T, [, ]R$ , we see 3 correct to-

<sup>4</sup>At this stage, we did not measure the initial agreement between the annotators since our goal was to achieve a high level of agreement in the course of the discussion. However, to follow the common practice in corpus annotation, we will provide in the near future the inter-annotator figures.

Transition		Condition
LEFT-BRACKET	$([\sigma i], B, V_c, [\zeta u'], E, \delta, o) \Rightarrow ([\sigma i], B, V_c, \cup\{o' \leftarrow (o + 1)'\}, [\zeta u' o'], E \cup \{(i, o'), (u', o')\}, o + 1)$	$i \neq 0$
RIGHT-BRACKET <sub>l</sub>	$([\sigma i], B, V_c, [\zeta u' o'], E, \delta, o) \Rightarrow ([\sigma i], B, V_c, [\zeta u'], E \cup \{(i, o')\}, \delta[o' \rightarrow l], o)$	$ \zeta  > 1$
SHIFT	$(\sigma, [i \beta], V_c, \zeta, E, \delta, o) \Rightarrow ([\sigma i], \beta, V_c, \zeta, E, \delta, o)$	
REDUCE	$([\sigma i], B, V_c, \zeta, E, \delta, o) \Rightarrow (\sigma, B, V_c, \zeta, E, \delta, o)$	$i \neq 0$

Figure 2: Possible parsing transitions.

1 SHIFT	$([0], [\text{He}, \text{believes}, \dots], \{0'\}, [0'], \{\}, \delta, 0) \Rightarrow ([0 \text{He}], [\text{believes}, \dots], \{0'\}, [0'], \{\}, \delta, 0)$
2 LEFT-BRACKET	$\Rightarrow ([0 \text{He}], [\text{believes}, \dots], \{0', 1'\}, [0' 1'], \{(\text{He}, 1'), (0', 1')\}, \delta, 1)$
3 RIGHT-BRACKET <sub>T</sub>	$\Rightarrow ([0 \text{He}], [\text{believes}, \dots], \{0', 1'\}, [0'], \{(\text{He}, 1')\}, \{(1' \rightarrow T)\}, 1)$
4 REDUCE	$\Rightarrow ([0], [\text{believes}, \text{in}, \dots], \{0', 1'\}, [0'], \{(\text{He}, 1'), (0', 1')\}, \{(1' \rightarrow T)\}, 1)$
5 SHIFT	$\Rightarrow ([0, \text{believes}], [\text{in}, \dots], \{0', 1'\}, [0'], \{(\text{He}, 1'), (0', 1')\}, \{(1' \rightarrow T)\}, 1)$
6 LEFT-BRACKET	$\Rightarrow ([0, \text{believes}], [\text{in}, \dots], \{0', 1', 2'\}, [0' 2'], \{(\text{He}, 1'), (0', 1'), (\text{believes}, 2'), (0', 2')\}, \{(1' \rightarrow T)\}, 2)$
7 SHIFT	$\Rightarrow ([\dots, \text{in}], [\text{what}, \dots], \{0', 1', 2'\}, [0' 2'], \{(\text{He}, 1'), (0', 1'), (\text{believes}, 2'), (0', 2')\}, \{(1' \rightarrow T)\}, 2)$
8 SHIFT	$\Rightarrow ([\dots, \text{in}, \text{what}], [\text{he}, \dots], \{0', 1', 2'\}, [0' 2'], \{(\text{He}, 1'), (0', 1'), (\text{believes}, 2'), (0', 2')\}, \{(1' \rightarrow T)\}, 2)$
9 LEFT-BRACKET	$\Rightarrow ([\dots, \text{in}, \text{what}], [\text{he}, \dots], \{0', 1', 2', 3'\}, [0' 2' 3'], \{\dots, (\text{believes}, 2'), (\text{what}, 3'), (2', 3')\}, \{(1' \rightarrow T)\}, 3)$
10 RIGHT-BRACKET <sub>T</sub>	$\Rightarrow ([\dots, \text{in}, \text{what}], [\text{he}, \dots], \{0', 1', 2', 3'\}, [0' 2'], \{\dots, (\text{what}, 3'), (2', 3')\}, \{(1' \rightarrow T), (3' \rightarrow T)\}, 3)$
11 REDUCE	$\Rightarrow ([\dots, \text{in}], [\text{he}, \dots], \{0', 1', 2', 3'\}, [0' 2'], \{\dots, (\text{what}, 3'), (2', 3')\}, \{(1' \rightarrow T), (3' \rightarrow T)\}, 3)$
12 SHIFT	$\Rightarrow ([\dots, \text{in}, \text{he}], [\text{plays}], \{0', 1', 2', 3'\}, [0' 2'], \{\dots, (\text{what}, 3'), (2', 3')\}, \{(1' \rightarrow T), (3' \rightarrow T)\}, 3)$
13 LEFT-BRACKET	$\Rightarrow ([\dots, \text{in}, \text{he}], [\text{plays}], \{0', 1', 2', 3', 4'\}, [0' 2' 4'], \{\dots, (2', 3'), (\text{he}, 4')\}, \{(1' \rightarrow T), (3' \rightarrow T)\}, 4)$
14 SHIFT	$\Rightarrow ([\dots, \text{in}, \text{he}, \text{plays}], [], \{0', 1', 2', 3', 4'\}, [0' 2' 4'], \{\dots, (2', 3'), (\text{he}, 4')\}, \{(1' \rightarrow T), (3' \rightarrow T)\}, 4)$
15 RIGHT-BRACKET <sub>R</sub>	$\Rightarrow ([0, \text{believes}, \text{in}, \text{he}, \text{plays}], [], \{0', 1', 2', 3', 4'\}, [0' 2'], \{\dots, (2', 4')\}, \{\dots, (4' \rightarrow R)\}, 4)$
16 RIGHT-BRACKET <sub>R</sub>	$\Rightarrow ([0, \text{believes}, \text{in}, \text{he}, \text{plays}], [], \{0', 1', 2', 3', 4'\}, [0'], \{\dots, (2', 4')\}, \{\dots, (4' \rightarrow R), (2' \rightarrow R)\}, 4)$
19 REDUCE <sub>.....</sub> REDUCE $\Rightarrow$	$([0], [], \{0', 1', 2', 3', 4'\}, [0'], \{\dots, (\text{he}, 4'), (2', 4'), (0', 2')\}, \{\dots, (4' \rightarrow R), (2' \rightarrow R)\}, 4)$

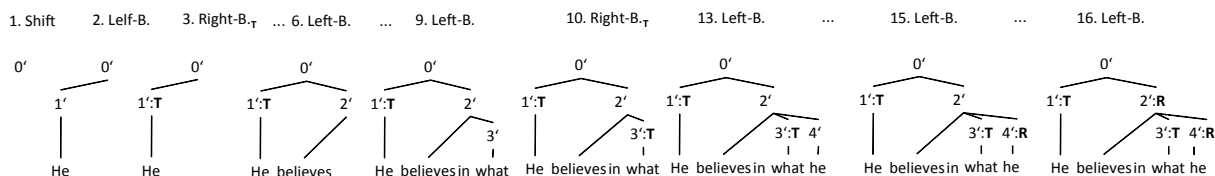


Figure 3: Transition sequence for the sentence: *He believes in what he plays.*

kens out of 5, i.e., an accuracy score of 60%. Note that the assignment  $] ] R$  instead of  $] R$  is considered as wrong. In contrast to this simple score, the labeled bracket score (LBS) and the unlabeled bracket score (UBS) consider the bracketing; the scores are calculated with the `evalb`-script as used for the evaluation of phrase structure parsers.

System	AS	LBS	UBS
sequence tagger	71.74	51.78	53.29
transition-based	88.67	68.95	74.33

Table 2: Accuracy scores for the assignment of the communicative labels.

## 5 Discussion and future work

The results of our experiments show that the interpretation of the annotation of PTB with IS as a transition-based parsing task is promising. Acceptable accuracy scores are achieved already with a very small training set. A direct comparison with other works on automatic annotation with IS, as e.g., (Postolache et al., 2005) with TFA, is not possible since the data sets and the annotation

schemata are different; see, e.g., (Hajičová, 2007) for a precise outline of the criteria for the annotation of TFA in the Prague school and a juxtaposition of TFA and the CommStr. However, it is instructive to observe that the AS we achieve with the transition parser is about the same as Postolache et al.’s accuracy with RIPPER and MAXENT models and only slightly below their performance with C4.5. This means that parsing is a valid alternative to token-oriented classification. However, we must be aware that parsing can only be applied if we assume the IS structure to be hierarchical (more precisely, a tree). The parser performance in terms of LBS and UBS, which capture the “bracketing” of the transitivity elements within the structure, are somewhat lower and can still be improved with a larger training sample (see below).

To have a clearer idea what the most recurrent mistakes of our IS parser are and whether they can be avoided (e.g., by a larger training sample) we carried out an error analysis of the resulting automatic annotation. This analysis has shown that sentences with clear (lexical, syntactic,

and/or punctuation) thematicity markers are analyzed correctly. Cf., e.g., (2) and (3):

(2) [*Indeed*]SP, [*the government*]T [*is taking a calculated risk*]R

(3) [*At the same time*]SP, [*the government*]T [*didn't want to appear to favor GM by allowing a minority stake [that]T [might preclude a full bid by Ford]*]R

In more complex sentences, the algorithm does not accurately detect the propositions involved, triggering the following errors: (a) consecutive themes (even if the second one begins with a verb) (4); (b) consecutive rhemes (even if there is no theme and there is no verb in the first rheme) (5); (c) reduced clauses are not labelled as embedded rhemes (6).

(4) [*In a prepared statement*]SP, [*GM*]T [*suggested its plans for Jaguar*]T [*would be more valuable in the long run than the initial windfalls investors*]T [*migh reap from a hostile Ford bid*]R.

(5) [*Erwin Tomash, the 67-year-old founder of this maker of data communications products and a former chairman and chief executive*]R, [*resigned as a director*]R.

(6) [*In national over-the-counter trading*]R, [*SFE technologies shares*]T [*closed yesterday at 31.25 cents a share, up 6.25 cents*]R.

Another detected error is that just the verb is labelled as rheme, which also brings embeddedness problems (7).

(7) “ [*Our intensive discussions with Jaguar , at their invitation*]R , ” [*GM said*]R , “ [*have as their objectives to create a cooperative business relationship with Jaguar [that]T [would]R provide for the continued independence of this great British car company...*]R.

Apart from these major errors, a number of minor errors can be detected—e.g., subordinate conjunctions are assumed as part of the theme if they are initial (8); initial locative or temporal specifiers are always labelled as part of the rheme (9); initial specifiers are confused with themes (10); etc.

(8) [*Although GM*]T [*has U.S. approval to buy up to 15% of Jaguar's stock*]SP, [*it*]T [*[hasn't yet disclosed how many shares it now owns]*]R.

(9) [*After a stronger - than - expected pace early this year*]R , [*analysts*]T [*say the market , after a series of sharp swings in recent months, now shows signs of retreating*]R.

(10) [*Under the circumstances*]T , [*Dataproducts said* , [*[Mr. Tomash]*]T [*said*]R]SP [*he*]T [*was un-*

*able to devote the time required because of other commitments*]R.

Finally, we found some few cases of non-annotated parts (*Dataproducts said* in (10)) or over-generated levels of embeddedness.

All these errors are likely to be straightened out with a larger training sample. The dependency curve between the accuracy ( $y$ -axis) and the size of the training set ( $x$ -axis) in Figure 4 shows that an increase of the size of the training set (e.g., to 1000 sentences) will further improve the scores. We are about to do this and apply the retrained transition-based analyser to the entire PTB. The information about how the resulting IS-bank can be accessed will be posted at <http://www.taln.upf.edu/resources>.

Our future work involves the extension of the annotation by other dimensions of the CommStr and a study of the correlation between the various dimensions of the IS and prosody. We assume that in particular the hierarchical structure of thematicity will correlate well with the prosodic structure of both simple and parenthetical or subordinate constructions, and thus contribute to a better quality in speech synthesis. Our positive experience with it in natural language text generation, where it guides syntactic realization, confirms this assumption.

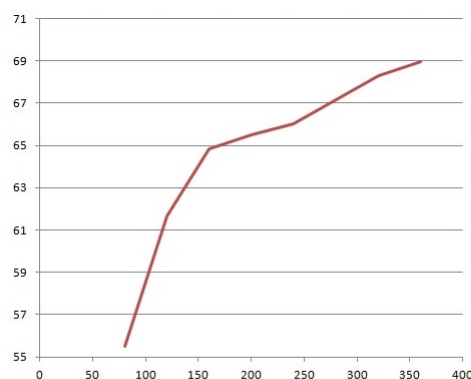


Figure 4: Dependency between the size of the training set and accuracy.

## References

- S. Calhoun, M. Nissim, M. Steedman and J. Brenier 2005. A Framework for Annotating Information Structure in Discourse. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 45–52. Ann Arbor, MA: Association for Corpus Linguistics.
- M. Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *EMNLP 2002.*, pages 1–8.

- M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*, pages 112–119.
- F. Daneš. 1970. Zur linguistischen Analyse der Textstruktur. *Folia Linguistica*, 4:72–78.
- S. Dipper, M. Götze, M. Stede, and T. Wegst. 2004. ANNIS: A Linguistic Database for Exploring Information Structure. In S. Ishihara, M. Schmitz and A. Schwarz (eds.). *Working Papers of the SFB 632, Interdisciplinary Studies on Information Structure 1*, pages 245–279. Potsdam: University of Potsdam.
- N. Erteschik-Shir. 2007. *Information Structure: The Syntax-Discourse Interface*. Oxford University Press, Oxford.
- J. Firbas. 1966. Non-thematic Subjects in Contemporary English. *Travaux Linguistiques de Prague*, 2:229–236.
- Jan Hajič, Massimiliano Ciaramita, Richard Johanson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CONLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 2009 CoNLL Shared Task*, pages 1–18.
- J. Hajič, J. Panevová, E. Hajicová, P. Sgall, P. Pajas, J. Štěpánek, J. Havelka, M. Mikulová, Z. Žabokrtský. 2006. *Prague Dependency Treebank 2.0*. Charles University Prague.
- E. Hajičová. 2007. The Position of TFA (Information Structure) in a Dependency Based Description of Language. In K. Gerdes, T. Reuther, and L. Wanner (eds.). *MTT 2007. Proceedings of the 3rd International Conference on Meaning-Text Theory*. Wiener Slavistischer Almanach, Sonderband 69, Munich & Vienna.
- E. Hajičová, B. Partee, P. Sgall. 1998. Topic-Focus Articulation, Tripartite Structures, and Semantic Content. Kluwer Academic Publishers, Dordrecht.
- I. Kruijff-Korbayová and M. Steedman. 2003. Discourse and Information Structure. *Journal of Logic, Language and Information*, 12(3):249–259.
- K. Lambrecht. 1994. *Information structure and sentence form: Topic, focus and the mental representations of discourse referents*. Cambridge University Press, Cambridge.
- M.P. Marcus, B. Santorini and M.A. Marcinkiewicz. . 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- V. Mathesius. 1929. Zur Satzperspektive im modernen Englisch. *Archiv für das Studium der neueren Sprachen und Literaturen*, 155:202–210.
- I.A. Mel'čuk. 2001. *Communicative Organization in Natural Language: The semantic-communicative structure of sentences*. Benjamins Academic Publishers, Amsterdam.
- J. Nivre, J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of CoNLL*, pages 49–56.
- J. Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL)*, pages 50–57.
- M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–105.
- P. Paggio 2006. Annotating Information Structure in a Corpus of Spoken Danish. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1606–1609.
- O. Postolache, I. Kruijff-Korbayová and G.-J. M. Kruijff 2005. Data-driven approaches for information structure identification. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 9–16.
- P. Sgall. Functional Sentence Perspective in a Generative Description of Language. 1967. *Prague Studies in Mathematical Linguistics*, 2:203–225.
- M. Steedman. Information Structure and the Syntax-Phonology Interface. 2000. *Linguistic Inquiry*, 31(4):649–685.
- M. Taulé, M.A. Martí and M. Recasens. 2008. AnCor: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*. Marrakesh, Morocco.
- C. Thielen, A. Schiller, S. Teufel and C. Stöckert. Guidelines für das Tagging deutscher Textkorpora mit STTS. 1999. Institute for Natural Language Processing, University of Stuttgart. <http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/>.
- E. Vallduví 1992. *Information Component*. Garland, New York and London.
- H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, pages 195–206.
- M. Zimmermann and C. Féry 2009. *Information Structure: Theoretical, Typological, and Experimental Perspectives*. Oxford University Press, Oxford.
- Y. Zhang and J. Nivre. 2011. Transition-based parsing with rich non-local features. In *Proceedings of ACL*.

# Constituency and Dependency Relationship from a Tree Adjoining Grammar and Abstract Categorical Grammar Perspective

Aleksandre Maskharashvili and Sylvain Pogodalla

INRIA, 54600 Villers-lès-Nancy, France

LORIA, UMR 7503, 54506 Vandœuvre-lès-Nancy, France

Sylvain.Pogodalla@inria.fr

Aleksandre.Maskharashvili@inria.fr

## Abstract

This paper gives an Abstract Categorical Grammar (ACG) account of (Kallmeyer and Kuhlmann, 2012)’s process of transformation of the derivation trees of Tree Adjoining Grammar (TAG) into dependency trees. We make explicit how the requirement of keeping a direct interpretation of dependency trees into strings results into lexical ambiguity. Since the ACG framework has already been used to provide a logical semantics from TAG derivation trees, we have a unified picture where derivation trees and dependency trees are related but independent equivalent ways to account for the same surface–meaning relation.

## 1 Introduction

Tree Adjoining Grammars (TAG) (Joshi et al., 1975; Joshi and Schabes, 1997) is a tree grammar formalism relying on two operations between trees: *substitution* and *adjunction*. In addition to the tree generated by a sequence of such operations, there is a *derivation tree* which records this sequence. Derivation trees soon appeared as good candidates to encode semantic-like relations between the elementary trees they glue together. However, some mismatch between these trees and the relative scoping of logical connectives and relational symbols, or between these trees and the dependency relations, have been observed. Solving these problems often leads to modifications of derivation tree structures (Schabes and Shieber, 1994; Kallmeyer, 2002; Joshi et al., 2003; Rambow et al., 2001; Chen-Main and Joshi, To appear).

While alternative proposals have succeeded in linking derivation trees to semantic representations using unification (Kallmeyer and Romero,

2004; Kallmeyer and Romero, 2007) or using an encoding (Pogodalla, 2004; Pogodalla, 2009) of TAG into the ACG framework (de Groote, 2001), only recently (Kallmeyer and Kuhlmann, 2012) has proposed a transformation from standard derivation trees to dependency trees.

This paper provides an ACG perspective on this transformation. The goal is twofold. First, it exhibits the underlying lexical blow up of the yield functions associated with the elementary trees in (Kallmeyer and Kuhlmann, 2012). Second, using the same framework as (Pogodalla, 2004; Pogodalla, 2009) allows us to have a shared perspective on a phrase-structure architecture and a dependency one and an equivalence on the surface-meaning relation they define.

## 2 Abstract Categorical Grammars

ACGs provide a framework in which several grammatical formalisms may be encoded (de Groote and Pogodalla, 2004). They generate languages of linear  $\lambda$ -terms, which generalize both string and tree languages. A key feature is to provide the user direct control over the parse structures of the grammar, the *abstract language*, which allows several grammatical formalisms to be defined in terms of ACG, in particular TAG (de Groote, 2002). We refer the reader to (de Groote, 2001; Pogodalla, 2009) for the details and introduce here only few relevant definitions and notations.

**Definition.** A higher-order linear signature is defined to be a triple  $\Sigma = \langle A, C, \tau \rangle$ , where:

- $A$  is a finite set of atomic types (also noted  $A_\Sigma$ ),
- $C$  is a finite set of constants (also noted  $C_\Sigma$ ),
- and  $\tau$  is a mapping from  $C$  to  $\mathcal{T}_A$  the set of types built on  $A$ :  $\mathcal{T}_A ::= A | \mathcal{T}_A \multimap \mathcal{T}_A$  (also noted  $\mathcal{T}_\Sigma$ ).

A higher-order linear signature will also be called

a vocabulary.  $\Lambda(\Sigma)$  is the set of  $\lambda$ -terms built on  $\Sigma$ , and for  $t \in \Lambda(\Sigma)$  and  $\alpha \in \mathcal{T}_\Sigma$  such that  $t$  has type  $\alpha$ , we note  $t :_\Sigma \alpha$  (the  $\Sigma$  subscript is omitted when it is obvious from the context).

**Definition.** An abstract categorial grammar is a quadruple  $\mathcal{G} = \langle \Sigma, \Xi, \mathcal{L}, s \rangle$  where:

1.  $\Sigma$  and  $\Xi$  are two higher-order linear signatures, which are called the abstract vocabulary and the object vocabulary, respectively;
2.  $\mathcal{L} : \Sigma \longrightarrow \Xi$  is a lexicon from the abstract vocabulary to the object vocabulary. It is a homomorphism<sup>1</sup> that maps types and terms built on  $\Sigma$  to types and terms built on  $\Xi$ . We note  $t :=_{\mathcal{G}} u$  if  $\mathcal{L}(t) = u$  and omit the  $\mathcal{G}$  subscript if obvious from the context.
3.  $s \in \mathcal{T}_\Sigma$  is a type of the abstract vocabulary, which is called the distinguished type of the grammar.

**Definition.** The abstract language of an ACG  $\mathcal{G} = \langle \Sigma, \Xi, \mathcal{L}, s \rangle$  is  $\mathcal{A}(\mathcal{G}) = \{t \in \Lambda(\Sigma) \mid t :_\Sigma s\}$

The object language of the grammar  $\mathcal{O}(\mathcal{G}) = \{t \in \Lambda(\Xi) \mid \exists u \in \mathcal{A}(\mathcal{G}). t = \mathcal{L}_G(u)\}$

Since there is no structural difference between the abstract and the object vocabulary as they both are higher-order signatures, ACGs can be combined in different ways. Either by having a same abstract vocabulary shared by several ACGs in order to make two object terms (for instance a string and a logical formula) share the same underlying structure as  $\mathcal{G}_{d-ed\ trees}$  and  $\mathcal{G}_{Log}$  in Fig. 1. Or by making the abstract vocabulary of an ACG the object vocabulary of another ACG, allowing the latter to control the admissible structures of the former, as  $\mathcal{G}_{yield}$  and  $\mathcal{G}_{d-ed\ trees}$  in Fig. 1.

### 3 TAG as ACG

As Fig. 1 shows, the encoding of TAG into ACG uses two ACGs  $\mathcal{G}_{d-ed\ trees} = \langle \Sigma_{der\theta}, \Sigma_{trees}, \mathcal{L}_{d-ed\ trees}, \mathbf{s} \rangle$  and  $\mathcal{G}_{yield} = \langle \Sigma_{trees}, \Sigma_{string}, \mathcal{L}_{yield}, \tau \rangle$ . We exemplify the encoding<sup>2</sup> of a TAG analyzing (1)<sup>3</sup>

<sup>1</sup>In addition to defining  $\mathcal{L}$  on the atomic types and on the constants of  $\Sigma$ , we have:

- If  $\alpha \multimap \beta \in \mathcal{T}_\Sigma$  then  $\mathcal{L}(\alpha \multimap \beta) = \mathcal{L}(\alpha) \multimap \mathcal{L}(\beta)$ .
- If  $x \in \Lambda(\Sigma)$  (resp.  $\lambda x.t \in \Lambda(\Sigma)$  and  $t u \in \Lambda(\Sigma)$ ) then  $\mathcal{L}(x) = x$  (resp.  $\mathcal{L}(\lambda x.t) = \lambda x.\mathcal{L}(t)$  and  $\mathcal{L}(t u) = \mathcal{L}(t)\mathcal{L}(u)$ )

with the proviso that for any constant  $c :_\Sigma \alpha$  of  $\Sigma$  we have  $\mathcal{L}(c) :=_\Xi \mathcal{L}(\alpha)$ .

<sup>2</sup>We refer the reader to (Pogodalla, 2009) for the details.

<sup>3</sup>The TAG literature typically uses this example, and (Kallmeyer and Kuhlmann, 2012) as well, to show the mismatch between the derivation trees and the expected se-

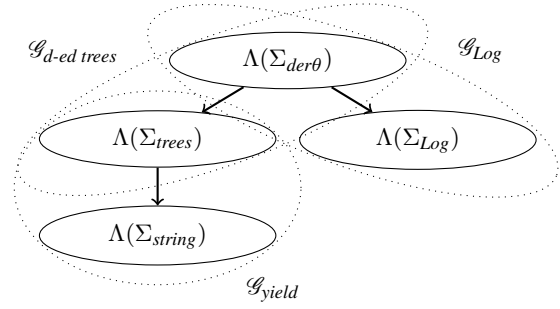


Figure 1: ACG architecture for TAG

- (1) John Bill claims Mary seems to love
- This sentence is usually analyzed in TAG with a derivation tree where the *to love* component scopes over all the other arguments, and where *claims* and *seems* are unrelated, as Fig. 2(a) shows.

The three higher-order signatures are:

$\Sigma_{der\theta}$ : Its atomic types include  $\mathbf{s}$ ,  $\mathbf{vp}$ ,  $\mathbf{np}$ ,  $\mathbf{s}_A$ ,  $\mathbf{vp}_A \dots$  where the  $X$  types stand for the categories  $X$  of the nodes where a substitution can occur while the  $X_A$  types stand for the categories  $X$  of the nodes where an adjunction can occur. For each elementary tree  $\gamma_{lex.\ entry}$  it contains a constant  $C_{lex.\ entry}$  whose type is based on the adjunction and substitution sites as Table 1 shows. It additionally contains constants  $I_X : X_A$  that are meant to provide a fake auxiliary tree on adjunction sites where no adjunction actually takes place in a TAG derivation.

$\Sigma_{trees}$ : Its unique atomic type is  $\tau$  the type of trees. Then, for any  $X$  of arity  $n$  belonging to the ranked alphabet describing the elementary trees of the TAG, we have a constant

$$X_n : \overbrace{\tau \multimap \dots \multimap \tau}^{n \text{ times}} \multimap \tau$$

$\Sigma_{string}$ : Its unique atomic type is  $\sigma$  the type of strings. The constants are the terminal symbols of the TAG (with type  $\sigma$ ), the concatenation  $+$  :  $\sigma \multimap \sigma \multimap \sigma$  and the empty string  $\varepsilon$  :  $\sigma$ .

Table 1 illustrates  $\mathcal{L}_{d-ed\ trees}$ .<sup>4</sup>  $\mathcal{L}_{yield}$  is defined as follows:

- $\mathcal{L}_{yield}(\tau) = \sigma$ ;
- for  $n > 0$ ,  $\mathcal{L}_{yield}(X_n) = \lambda x_1 \dots x_n. x_1 + \dots + x_n$ ;
- for  $n = 0$ ,  $X_0 : \tau$  represents a terminal sym-

mantics and the relative scopes of the predicates.

<sup>4</sup>With  $\mathcal{L}_{d-ed\ trees}(X_A) = \tau \multimap \tau$  and for any other type  $X$ ,  $\mathcal{L}_{d-ed\ trees}(X_A) = \tau$ .

$\text{bol}$  and  $\mathcal{L}_{\text{yield}}(X_0) = X$ .

Then, the derivation tree, the derived tree, and the yield of Fig. 2 are represented by:

$$\begin{aligned} t_0 &= C_{\text{to love}} (C_{\text{claims}} I_{\mathbf{s}} C_{\text{Bill}}) (C_{\text{seems}} I_{\mathbf{vp}}) C_{\text{Mary}} C_{\text{John}} \\ \mathcal{L}_{\text{d-ed trees}}(t_0) &= \mathbf{s}_2 (\text{np}_1 \text{ John}) (\mathbf{s}_2 (\text{np}_1 \text{ Bill}) (\mathbf{vp}_2 \text{ claims} (\mathbf{s}_2 \\ &\quad (\text{np}_1 \text{ Mary}) (\mathbf{vp}_2 \text{ seems} (\mathbf{vp}_1 \text{ to love})))) \\ \mathcal{L}_{\text{yield}}(\mathcal{L}_{\text{d-ed trees}}(t_0)) &= \text{John} + \text{Bill} + \text{claims} \\ &\quad + \text{Mary} + \text{seems} + \text{to love} \end{aligned}$$

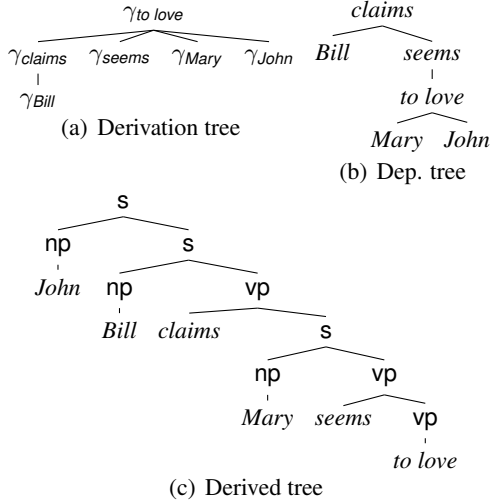


Figure 2: *John Bill claims Mary seems to love*

## 4 From Derivation Trees to Dependency Trees

(Kallmeyer and Kuhlmann, 2012)’s process to translate derivation trees into dependency trees is a two-step process. The first one does the actual transformation, using macro-tree transduction, while the second one modifies the way to get the yield from the dependency trees rather than from the derivation ones.

### 4.1 From Derivation To Dependency Trees

This transformation aims at modeling the differences in scope of the argument between the derivation tree for (1) shown in Fig. 2(a) and the corresponding dependency tree shown in Fig. 2(b). For instance, in the derivation trees, *claims* and *seems* are under the scope of *to love* while in the dependency tree this order is reversed. According to (Kallmeyer and Kuhlmann, 2012), such edge reversal is due to the fact that an edge between a *complement taking adjunction* (CTA) and an initial tree has to be reversed, while the other edges remain unchanged.

Moreover, in case an initial tree accepts several adjunction of CTAs, (Kallmeyer and Kuhlmann, 2012) hypothesizes that the farther from the head a CTA is, the higher it is in the dependency tree. In the case of *to love*, the  $\mathbf{s}$  node is farther from the head than the  $\mathbf{vp}$  node. Therefore any adjunction on the  $\mathbf{s}$  node (e.g. *claims*) should be higher than the one on the  $\mathbf{vp}$  node (e.g. *seems*) in the dependency tree. We represent the dependency tree for (1) as  $t'_0 = d_{\text{claims}} d_{\text{Bill}} (d_{\text{seems}} (d_{\text{to love}} d_{\text{John}} d_{\text{Mary}}))$ .

In order to do such reversing operations, (Kallmeyer and Kuhlmann, 2012) uses Macro Tree Transducers (MTTs) (Engelfriet and Vogler, 1985). Note that the MTTs they use are *linear*, i.e. non-copying. It means that any node of an input tree cannot be translated more than once. (Yoshinaka, 2006) has shown how to encode such MTTs as the composition  $\mathcal{G}' \circ \mathcal{G}^{-1}$  of two ACGs, and we will use a very similar construct.

### 4.2 The Yield Functions

(Kallmeyer and Kuhlmann, 2012) adds to the transformation from derivation trees to dependency trees the additional constraint that the string associated with a dependency structure is computed directly from the latter, without any reference to the derivation tree. To achieve this, they use two distinct yield functions:  $\text{yield}_{\text{TAG}}$  from derivation trees to strings, and  $\text{yield}_{\text{dep}}$  from dependency trees to strings.

Let us imagine an initial tree  $\gamma_i$  and an auxiliary tree  $\gamma_a$  with no substitution nodes. The yield of the derived tree resulting from the operations of the derivation tree  $\gamma$  of Fig. 3 defined in (Kallmeyer and Kuhlmann, 2012) is such that

$$\begin{aligned} \text{yield}_{\text{TAG}}(\gamma) &= a_1 + w_1 + a_2 + w_2 + a_3 \\ &= (\text{yield}_{\text{TAG}}(\gamma_i))(\text{yield}_{\text{TAG}}(\gamma_a)) \\ &= (\lambda \langle x_1, x_2 \rangle. a_1 + x_1 + a_2 + x_2 \\ &\quad + a_3) \langle w_1, w_2 \rangle \end{aligned}$$

where  $\langle x, y \rangle$  denotes a tuple of strings.

Because of the adjunction, the corresponding dependency structure has a reverse order ( $\gamma' = \gamma'_a(\gamma'_i)$ ), the requirement on  $\text{yield}_{\text{dep}}$  imposes that

$$\begin{aligned} \text{yield}_{\text{dep}}(\gamma') &= a_1 + w_1 + a_2 + w_2 + a_3 \\ &= (\text{yield}_{\text{dep}}(\gamma'_a))(\text{yield}_{\text{dep}}(\gamma'_i)) \\ &= (\lambda \langle x_1, x_2, x_3 \rangle. x_1 + w_1 + x_2 + w_2 \\ &\quad + x_3) \langle a_1, a_2, a_3 \rangle \end{aligned}$$

In the interpretation of derivation trees as strings, initial trees (with no substitution nodes)



Abstract constants of $\Sigma_{der\theta}$	Their images by $\Sigma_{der\theta}$	The corresponding TAG trees
$C_{John} : np$	$c_{John} : \tau$ $= np_1 John$	$\gamma_{John} =$ 
$C_{seems} : vp_A \multimap vp_A$	$c_{seems} : (\tau \multimap \tau) \multimap (\tau \multimap \tau)$ $= \lambda^0 vx.v (vp_2 seems x)$	$\gamma_{seems} =$ 
$C_{to\ love} : s_A \multimap vp_A \multimap np$ $\multimap np \multimap s$	$c_{to\ love} : (\tau \multimap \tau) \multimap (\tau \multimap \tau) \multimap \tau \multimap \tau \multimap \tau$ $= \lambda^0 avso.s_2 o$ $(a (s_2 s (v (vp_1 to\ love))))$	$\gamma_{to\ love} =$ 
$C_{claims} : s_A \multimap vp_A$ $\multimap np \multimap s_A$	$c_{claims} : (\tau \multimap \tau) \multimap (\tau \multimap \tau) \multimap \tau \multimap \tau \multimap \tau$ $= \lambda^0 avsc.a (s_2 s (a (vp_2 claims c)))$	$\gamma_{claims} =$ 
$I_X : X_A$	$\lambda x.x : \tau \multimap \tau$	

Table 1: TAG as ACG: the  $\mathcal{L}_{d-ed\ trees}$  lexicon

are interpreted as functions from tuples of strings into strings, and auxiliary trees as tuples of strings. The interpretation of dependency trees as strings leads us to interpret initial trees as tuples of strings and auxiliary trees as function from tuples of strings to strings.

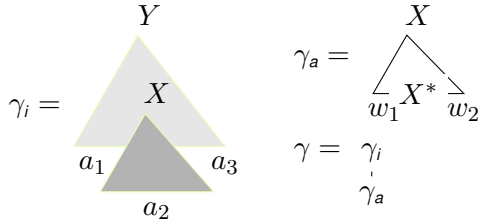


Figure 3: Yield from derivation trees

Indeed, an initial tree can have several adjunction sites. In this case, to be ready for another adjunction after a first one, the first result itself should be a tuple of strings. So an initial tree (with no substitution nodes) with  $n$  adjunction sites is interpreted as a  $(2n + 1)$ -tuple of strings. Accordingly, depending on the location where it can adjoint, an auxiliary tree is interpreted as a function from  $(2k + 1)$ -tuple of strings to  $(2k - 1)$ -tuple of strings.

Taking into account that to model trees having the substitution nodes is then just a matter of adding  $k$  string parameters where  $k$  is the number of substitution nodes in a tree. Then using the interpretation:

$$\begin{aligned}
 yield_{dep}(d_{to\ love}) &= \lambda x_{11} x_{21}. \langle x_{11}, x_{21}, to\ love, \varepsilon, \varepsilon \rangle \\
 yield_{dep}(d_{seems}) &= \lambda \langle x_{11}, x_{12}, x_{13}, x_{14}, x_{15} \rangle. \\
 &\quad \langle x_{11}, x_{12} + seems + x_{13}x_{14}, x_{15} \rangle \\
 yield_{dep}(d_{claims}) &= \lambda x_{21} \langle x_{11}, x_{13}, x_{14} \rangle. \\
 &\quad \langle x_{11} + x_{21} + claims + x_{14} + x_{13} \rangle
 \end{aligned}$$

we can check that

$$yield_{dep}(d_{claims} d_{Bill} (d_{seems}(d_{to\ love} d_{Mary} d_{John}))) = \langle John + Bill + claims + Mary + seems + to\ love \rangle$$

**Remark.** The given interpretation of  $d_{to\ love}$  is only valid for structures reflecting adjunctions both on the  $s$  node and on the  $vp$  node of  $\gamma_{to\ love}$ . So actually, an initial tree such as  $\gamma_{to\ love}$  yields four interpretations: one with the two adjunctions (5-tuple), two with one adjunction either on the  $vp$  node or on the  $s$  node (3-tuple), and one with no adjunction (1-tuple). The two first cases correspond to the sentences (2a) and (2b).<sup>5</sup> Accordingly, we need multiple interpretations for the auxiliary trees, for instance for the two occurrences of *seems* in (3) where the yield of the last one  $yield_{dep}(d_{seems})$  maps a 5-tuple to a 3-tuple, and the yield of the first one maps a 3-tuple to a 3-tuple. And  $yield_{dep}(d_{claims})$  maps a 3-tuple to a 1-tuple of strings. We will mimic this behavior by introducing as many different non-terminal symbols for the dependency structures in our ACG setting.

- (2) a. John Bill claims Mary seems to love  
b. John Mary seems to love
- (3) John Bill seems to claim Mary seems to love

**Remark.** Were we not interested in the yields but only in the dependency structures, we wouldn't have to manage this ambiguity. This is true both for (Kallmeyer and Kuhlmann, 2012)'s approach and ours. But as we have here a unified framework for the two-step process they propose, this lexical blow up will result in a multiplicity of types as Section 5 shows.

<sup>5</sup>As the two other ones are not correct English sentences, we can rule them out. However, from a general perspective, we should take such cases into account.

## 5 Disambiguated Derivation Trees

In order to encode the MTT acting on derivation trees, we introduce a new *abstract* vocabulary  $\Sigma'_{der\theta}$  for *disambiguated derivation trees* as in (Yoshinaka, 2006). Instead of having only one constant for each initial tree as in  $\Sigma_{der\theta}$ , we have as many of them as adjunction combinations. For instance,  $\gamma_{to\ love}$  gives rise to the several constants in  $\Sigma'_{der\theta}$ :

$$\begin{aligned} C_{to\ love}^{11} &: \mathbf{s}_A^{31} \multimap \mathbf{vp}_A^{53} \multimap \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s} \\ C_{to\ love}^{10} &: \mathbf{s}_A^{31} \multimap \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s} \\ C_{to\ love}^{01} &: \mathbf{vp}_A^{31} \multimap \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s} \\ C_{to\ love}^{00} &: \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s} \end{aligned}$$

Here,  $C_{to\ love}^{11}$  is used to model sentences where both adjunctions are performed into  $\gamma_{to\ love}$ .  $C_{to\ love}^{10}$  and  $C_{to\ love}^{01}$  are used for sentences where only one adjunction at the **s** or at the **vp** node occurs respectively.  $C_{to\ love}^{00} : \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s}$  is used when no adjunction occurs.<sup>6</sup> This really mimics (Yoshinaka, 2006)'s encoding of (Kallmeyer and Kuhlmann, 2012) MTT rules:

$$\begin{aligned} \langle q_0, C_{to\ love}(x_1, x_2, x_3, x_4) \rangle &\rightarrow \\ \langle q_2, x_2 \rangle (\langle q_4, x_4 \rangle (d_{to\ love}(\langle q_1, x_1 \rangle, \langle q_3, x_3 \rangle))) & \\ \langle q_0, C_{to\ love}(x_1, x_2, x_3) \rangle &\rightarrow \\ \langle q_2, x_2 \rangle (d_{to\ love}(\langle q_1, x_1 \rangle, \langle q_3, x_3 \rangle)) & \\ \langle q_0, C_{to\ love}(x_1, x_2, x_3) \rangle &\rightarrow \\ \langle q_4, x_4 \rangle (d_{to\ love}(\langle q_1, x_1 \rangle, \langle q_3, x_3 \rangle)) & \\ \langle q_0, C_{to\ love}(x_1, x_2) \rangle &\rightarrow \\ d_{to\ love}(\langle q_1, x_1 \rangle, \langle q_3, x_3 \rangle) & \end{aligned}$$

where the states  $q_0, q_1, q_2, q_3$  and  $q_4$  are given the names **s**, **np**,  $\mathbf{s}_A^{31}$ , **np**, and  $\mathbf{vp}_A^{53}$  resp.

Moreover,  $\mathbf{s}_A^{31}, \mathbf{vp}_A^{31}, \dots, \mathbf{vp}_A^{2(n+1)2(n-1)}$  ... are designed in order to indicate that a given adjunction has  $n$  adjunctions above it (i.e. which scope over it). The superscripts  $(2(n+1))(2(n-1))$  express that an adjunction that has  $n$  adjunctions above it is translated as a function that takes a  $2(n+1)$ -tuple as argument and returns a  $2(n-1)$ -tuple.

To model auxiliary trees which are CTAs we need a different strategy. For each such adjunction tree  $T$  we have two sets in  $\Sigma'_{der\theta}$ :  $\mathcal{S}_T^1$  the set of constants which can be adjoined into **initial** trees and  $\mathcal{S}_T^2$  the set of constants which can be adjoined into **auxiliary** trees.

For instance,  $\gamma_{seems}$  would generate  $\mathcal{S}_{seems}^1$  that includes  $C_{seems31}^{11}, C_{seems31}^{10}, C_{seems31}^{01}, C_{seems31}^{00}, C_{seems53}^{11}$  etc.  $C_{seems31}^{00}$  is of type  $\mathbf{vp}_A^{31}$ , which means that it can be adjoined into initial trees which contain  $\mathbf{vp}_A^{31}$  as its argument type (e.g.  $C_{to\ love}^{01}$ ).

<sup>6</sup>See note 5.

$C_{seems31}^{11}$  is of type  $\mathbf{s}_A^{3-3} \multimap \mathbf{vp}_A^{3-3} \multimap \mathbf{vp}_A^{31}$ . It means it expects two adjunctions at its **s** and **vp** nodes respectively and returns back a term of type  $\mathbf{vp}_A^{31}$  (as in *John claims to appear to seem to love Mary*). Here,  $\mathbf{s}_A^{3-3}$  and  $\mathbf{vp}_A^{3-3}$  are types used for modeling *adjunction on adjunctions*.

When an auxiliary tree is adjoined into another auxiliary tree as in (3), we do not allow the former to modify the tupleness of the latter. For instance  $\gamma_{seems}$  would generate  $\mathcal{S}_{seems}^2$  that includes  $C_{seems3-3}^{11}, C_{seems3-3}^{10}, C_{seems3-3}^{01}, C_{seems3-3}^{00}, C_{seems5-5}^{11}$  etc.  $C_{seems3-3}^{00}$  has a subscript  $(k-k)$  that correspond to adjunctions into adjunction trees. The type of  $C_{seems3-3}^{00}$  is  $\mathbf{vp}_A^{3-3}$ , meaning that it can directly adjoin into auxiliary trees which have arguments of type  $\mathbf{vp}_A^{3-3}$ .  $C_{seems3-3}^{01}$  is of type  $\mathbf{vp}_A^{3-3} \multimap \mathbf{vp}_A^{3-3}$ , which means that it itself expects an adjunction and the result can be adjoined into another adjunction tree.

Now it is easy to define  $\mathcal{L}_{der}$  from  $\Sigma'_{der\theta}$  to  $\Sigma_{der\theta}$ . It maps every type  $X \in \Sigma'_{der\theta}$  to  $X \in \Sigma_{der\theta}$  and every  $X_A^N$  to  $X_A$ ; types without numbers are mapped to themselves, i.e. **s** to **s**, **np** to **np**, etc. Moreover, the different *versions* of some constant, that were introduced in order to extract the yield, are translated using only one constant and *fake* adjunctions. For instance:

$$\begin{aligned} \mathcal{L}_{der}(C_{to\ love}^{11}) &= C_{to\ love} \\ \mathcal{L}_{der}(C_{to\ love}^{10}) &= \lambda x s o. C_{to\ love} x I_{vp} s o \\ \mathcal{L}_{der}(C_{to\ love}^{00}) &= C_{to\ love} I_s I_{vp} \end{aligned}$$

## 6 Encoding a Dependency Grammar

The ACG of (Pogodalla, 2009) mapping TAG derivation trees to logical formulas already encoded some reversal of the predicate-argument structure. Here we map the disambiguated derivation trees to dependency structures. The vocabulary that define these *dependency trees* is  $\Sigma_{dep}$ . It is also designed to allow us to build two lexicons from it to  $\Sigma_{string}$  (to provide a direct yield function) and to  $\Sigma_{Log}$  (to provide a logical semantic representation).

In  $\Sigma_{dep}$  constants are typed as follows:  $d_{to\ love}^5 : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^5$ . Here,  $\tau_{np}^1$  is the type into which the **np** type is translated from *disambiguated derivation tree*. The superscript 1 indicates that  $\tau_{np}^1$  will be translated into 1-tuple into  $\Sigma_{string}$ . Now, it is easy to see that in order to translate  $C_{to\ love}^{10} : \mathbf{s}_a^{31} \multimap \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s}$  and  $C_{to\ love}^{01} : \mathbf{vp}_a^{31} \multimap \mathbf{np} \multimap \mathbf{np} \multimap \mathbf{s}$ , we need to

have constants like:  $d_{to\ love}^3 \mathbf{S} : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^3$  and  $d_{to\ love}^3 \mathbf{V} : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^3$ .

Moreover, we have constants for adjunction trees, like  $d_{seems}^5 : \tau^5 \multimap \tau^3$  that will be used in the translation of  $C_{seems53}^{01}$ , and  $d_{seems}^{5-5} : \tau^5 \multimap \tau^5$  for  $C_{seems5-5}^{00}$ . Furthermore, additional constants are needed to have things *correctly* typed. For this reason, the constants  $d_3^1, d_5^3$  etc. are introduced. Each  $d_{2n+1}^{2n-1}$  has type  $\tau^{2n+1} \multimap \tau^{2n-1}$ .

Finally, non-CTAs like  $n_A, n_A^d, vp_A$  and  $s_A$  are translated as  $\tau_{n_A}^2, \tau_{n_A^d}^2, \tau_{vp}^2$ , and  $\tau_s^2$  respectively. A superscript 2 indicates that they are modeled as 2-tuples in  $\Sigma_{string}$ .

Now we can define  $\mathcal{L}_{dep}$ , the lexicon from  $\Sigma'_{der\theta}$  to  $\Sigma_{dep}$  translating disambiguated derivation trees into dependency trees:

$$\begin{aligned} \mathcal{L}_{dep}(\mathbf{s}) &= \tau^1 \\ \mathcal{L}_{dep}(\mathbf{np}) &= \tau_{np}^1 \\ \mathcal{L}_{dep}(X_A^{(2n+1)(2n-1)}) &= \tau^{2n+1} \multimap \tau^{2n-1} \\ \mathcal{L}_{dep}(X_A^{(2n+1)(2n+1)}) &= \tau^{2n+1} \multimap \tau^{2n+1} \\ &\text{for } X \in \mathbf{s}_A^{31}, \mathbf{vp}_A^{53} \dots \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{dep}(C_{to\ love}^{11}) &= \lambda S V s o. S(V(d_{to\ love}^5 s o)) \\ \mathcal{L}_{dep}(C_{to\ love}^{10}) &= \lambda S s o. S(d_{to\ love}^3 s o) \\ \mathcal{L}_{dep}(C_{to\ love}^{01}) &= \lambda S V s o. V(d_{to\ love}^3 s o) \\ \mathcal{L}_{dep}(C_{seems53}^{00}) &= \lambda x. d_{seems}^5 x \\ \mathcal{L}_{dep}(C_{seems53}^{01}) &= \lambda V x. d_{53}^5(V(d_{seems}^5 x)) \\ \mathcal{L}_{dep}(C_{seems53}^{11}) &= \lambda S V x. d_{53}^5(S(V(d_{seems}^5 x))) \\ \mathcal{L}_{dep}(C_{seems5-5}^{01}) &= \lambda V x. V(d_{seems}^{5-5} x) \\ \mathcal{L}_{dep}(C_{seems5-5}^{00}) &= \lambda x. d_{seems}^{5-5} x \end{aligned}$$

Furthermore, we describe  $\Sigma_{Log}$ <sup>7</sup> and define two lexicons:  $\mathcal{L}_{dep. yield} : \Sigma_{dep} \longrightarrow \Sigma_{string}$  and  $\mathcal{L}_{dep. log} : \Sigma_{dep} \longrightarrow \Sigma_{Log}$ . Table 2 provides examples of these two translations.

$\mathcal{L}_{dep. yield}$ : It translates any atomic type  $\tau^n$  or  $\tau_X^n$  with  $X \in \{n_A, n_A^d \dots\}$  as a  $n$ -tuple of string  $(\underbrace{\sigma \multimap \sigma \dots \multimap \sigma}_{n+1\text{-times}}) \multimap \sigma$ .<sup>8</sup>

$\Sigma_{Log}$ : Its atomic types are  $e$  and  $t$  and we have the constants: **john**, **mary**, **bill** of type  $e$ , the constant **love** of type  $e \multimap e \multimap t$ , the constant **claim** of type  $e \multimap t \multimap t$  and the constant **seem** of type  $t \multimap t$ .

$\mathcal{L}_{dep. log}$ : Each  $\tau^{2(n+1)}$  is mapped to  $t$ ,  $\tau_{np}^1$  is mapped to  $(e \multimap t) \multimap t$ ,  $\tau_{n_A^d}^2$  is mapped to  $(e \multimap t) \multimap (e \multimap t) \multimap t$ . The types

<sup>7</sup>We refer the reader to (Pogodalla, 2009) for the details.

<sup>8</sup>We encode a  $n$ -tuple  $\langle M_1, \dots, M_n \rangle$  as  $\lambda f. f M_1 M_2 \dots M_n$  where each  $M_i$  has type  $\sigma$ .

of non-complement-taking verbal or sentential adjuncts  $\tau_{vp}^2$  and  $\tau_s^2$  are translated as  $t \multimap t$ .

Let us show for the sentence (1) how the ACGs defined above work with the data provided in Table 2. Its representation in  $\Sigma'_{der\theta}$  is:  $T_0 = C_{to\ love}^{11} (C_{claims31} C_{Bill}) C_{seems53} C_{Mary} C_{John}$ . Then

$$\mathcal{L}_{der}(T_0) = t_0$$

and

$$\mathcal{L}_{dep}(T_0) = d_{claims}^1 d_{Bill} (d_{seems}^{53} (d_{to\ love}^5 d_{Mary} d_{John})) = t'_0$$

and finally

$$\begin{aligned} \mathcal{L}_{dep. yield}(t'_0) &= \mathcal{L}_{yield}(\mathcal{L}_{d-ed\ trees}(t_0)) \\ &= \lambda f. f(\mathbf{John} + (\mathbf{Bill} + (\mathbf{claims} \\ &\quad + ((\mathbf{Mary} + ((\mathbf{seems} + \mathbf{to\ love}) + \epsilon)) + \epsilon)))) \end{aligned}$$

and

$$\mathcal{L}_{dep. log}(t'_0) = \mathbf{claim\ bill\ (seem\ (love\ john\ mary))}$$

## 7 Conclusion

In this paper, we have given an ACG perspective on the transformation of the derivation trees of TAG to the dependency trees proposed in (Kallmeyer and Kuhlmann, 2012). Figure 4 illustrates the architecture we propose. This transformation is a two-step process using first a macro-tree transduction then an interpretation of dependency trees as (tuples of) strings. It was known from (Yoshinaka, 2006) how to encode a macro-tree transducer into a  $\mathcal{L}_{dep} \circ \mathcal{L}_{der}^{-1}$  ACG composition. Dealing with typed trees to represent derivation trees allows us to provide a meaningful (wrt. the TAG formalism) abstract vocabulary  $\Sigma'_{der\theta}$  encoding this macro-tree transducer. The encoding of the second step then made explicit the lexical blow up for the interpretation of the functional symbols of the dependency trees in (Kallmeyer and Kuhlmann, 2012)'s construct. It also provides a push out (in the categorical sense) of the two morphisms from the disambiguated derivation trees to the derived trees and to the dependency trees. The diagram is completed with the yield function from the derived trees and from the dependency trees to the string vocabulary.

Finally, under the assumption of (Kallmeyer and Kuhlmann, 2012) of plausible dependency structures, we get two possible grammatical approaches to the surface-semantics relation that are related but independent: it can be equivalently modeled using either a phrase structure or a dependency model.

Abstract constants of $\Sigma_{dep}$	Their images by $\mathcal{L}_{dep. yield}$	Their images by $\mathcal{L}_{dep. log}$
$d_{John} : \tau_{np}^1$	<i>John</i>	$\lambda P.P \text{ john}$
$d_{to\ love}^5 : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^5$	$\lambda S O.\lambda f.f O S \text{ to love } \epsilon \in$	$\lambda O S.S(\lambda x.O(\lambda y.(\mathbf{love} x y)))$
$d_{to\ love}^{3s} : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^3$	$\lambda S O.\lambda f.f O (S + \text{to love}) \epsilon$	$\lambda O S.S(\lambda x.O(\lambda y.(\mathbf{love} x y)))$
$d_{to\ love}^{3v} : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^3$	$\lambda S O.\lambda f.f (O + S) \text{ to love } \epsilon$	$\lambda O S.S(\lambda x.O(\lambda y.(\mathbf{love} x y)))$
$d_{claims}^1 : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^1$	$\lambda S c.\lambda f.c(\lambda x_1 x_2 x_3.f(x_1 + S + \mathbf{claims} + x_2 + x_3))$	$\lambda S c.S(\lambda x.\mathbf{claim} x c)$
$d_{claims}^3 : \tau_{np}^1 \multimap \tau_{np}^1 \multimap \tau^3$	$\lambda S c.\lambda f.c(\lambda x_1 x_2 x_3.f(x_1 + S) (\mathbf{claims} + x_2 + x_3))$	$\lambda S c.S(\lambda x.\mathbf{claim} x c)$
$d_{seems}^{5-5} : \tau^5 \multimap \tau^5$	$\lambda c.\lambda f.c(\lambda x_1 \dots x_5.f x_1 (x_2 + \mathbf{seems} + x_3 + x_4)x_5)$	$\lambda c.\mathbf{seem} c$
$d_{seems}^5 : \tau^5 \multimap \tau^3$	$\lambda c.\lambda f.c(\lambda x_1 \dots x_5.f x_1 x_2 (\mathbf{seems} + x_3) x_4 x_5)$	$\lambda c.\mathbf{seem} c$
$d_{2n+1}^{2n-1} : \tau^{2n+1} \multimap \tau^{2n-1}$	$\lambda g f.g(\lambda x_1 \dots x_n.f x_1 \dots (x_n + x_{n+1} + x_{n+2}) \dots x_{2n+1})$	$\lambda x.x : t \multimap t$

Table 2: Lexicons for yield and semantics from the dependency vocabulary

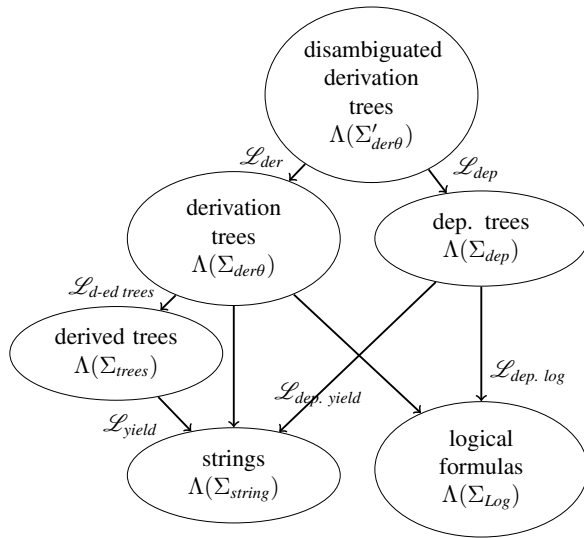


Figure 4: General architecture

## References

Joan Chen-Main and Aravind K. Joshi. To appear. A dependency perspective on the adequacy of tree local multi-component tree adjoining grammar. *Journal of Logic and Computation*.

Philippe de Groote and Sylvain Pogodalla. 2004. On the expressive power of Abstract Categorical Grammars: Representing context-free formalisms. *Journal of Logic, Language and Information*, 13(4):421–438.

Philippe de Groote. 2001. Towards abstract categorical grammars. In *Proceedings of ACL*, pages 148–155.

Philippe de Groote. 2002. Tree-adjoining grammars as abstract categorical grammars. In *Proceedings of TAG+6*, pages 145–150. Università di Venezia.

Joost Engelfriet and Heiko Vogler. 1985. Macro tree transducers. *J. Comput. Syst. Sci.*, 31(1):71–146.

Aravin K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume 3, chapter 2. Springer.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163.

Aravind K. Joshi, Laura Kallmeyer, and Maribel Romero. 2003. Flexible composition in Itag: Quantifier scope and inverse linking. In Harry Bunt, Ielka van der Sluis, and Roser Morante, editors, *Proceedings of IWCS-5*.

Laura Kallmeyer and Marco Kuhlmann. 2012. A formal model for plausible dependencies in lexicalized tree adjoining grammar. In *Proceedings of TAG+11*, pages 108–116.

Laura Kallmeyer and Maribel Romero. 2004. Itag semantics with semantic unification. In *Proceedings of TAG+7*, pages 155–162.

Laura Kallmeyer and Maribel Romero. 2007. Scope and situation binding for Itag. *Research on Language and Computation*, 6(1):3–52.

Laura Kallmeyer. 2002. Using an enriched tag derivation structure as basis for semantics. In *Proceedings of TAG+6*.

Sylvain Pogodalla. 2004. Computing Semantic Representation: Towards ACG Abstract Terms as Derivation Trees. In *Proceedings of TAG+7*, pages 64–71, Vancouver, BC, Canada. <http://hal.inria.fr/inria-00107768>.

Sylvain Pogodalla. 2009. Advances in Abstract Categorical Grammars: Language Theory and Linguistic Modeling. ESSLLI 2009 Lecture Notes, Part II. <http://hal.inria.fr/hal-00749297>.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-Substitution Grammars. *Computational Linguistics*.

Yves Schabes and Stuart M. Shieber. 1994. An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20(1):91–124.

Ryo Yoshinaka. 2006. *Extensions and Restrictions of Abstract Categorical Grammars*. Phd, University of Tokyo.

# Named Entity Extraction Using Information Distance\*

Sangameshwar Patil<sup>†</sup>, Sachin Pawar<sup>‡</sup>, Girish K. Palshikar

Tata Research Development and Design Centre, TCS

54B, Hadapsar Industrial Estate, Pune 411013, India

{sangameshwar.patil, sachin7.p, gk.palshikar}@tcs.com

## Abstract

Named entities (NE) are important information carrying units within documents. *Named Entity extraction (NEX)* task consists of automatic construction of a list of phrases belonging to each NE of interest. NEX is important for domains which lack a corpus with tagged NEs. We present an enhanced version and improved results of our unsupervised (bootstrapping) NEX technique (Patil et al., 2013) and establish its domain independence using experimental results on corpora from two different domains: agriculture and mechanical engineering (IC engine<sup>1</sup> parts). We use a new variant of Multiword Expression Distance (MED) (Bu et al., 2010) to quantify proximity of a candidate phrase with a given NE type. MED itself is an approximation of the information distance (Bennett et al., 1998). Efficacy of our method is shown using experimental comparison with pointwise mutual information (PMI), BASILISK and KNOWITALL. Our method discovered 8 new plant diseases which are not found in Wikipedia. To the best of our knowledge, this is the first use of NEX techniques for agriculture and mechanical engineering (engine parts) domains.

## 1 Introduction

Agriculture is an activity of fundamental importance in all societies. For example, agriculture plays a vital role in the economy of India: generating second largest farm output in the world,

\*Preliminary version of this paper was presented as a poster at NLDB 2013.

<sup>†</sup> Doctoral research scholar at Dept. of CSE, IIT Madras

<sup>‡</sup> Doctoral research scholar at Dept. of CSE, IIT Bombay

<sup>1</sup> Internal Combustion engine

providing 52% of rural employment ( $\approx$  250 million people) and contributing  $\approx$  15% to the GDP<sup>2</sup>. Hence timely and widespread dissemination of agriculture-related information is important. Such information - extracted, collated and summarized from a variety of document sources such as news, reports, web-sites and scientific literature - can be used to improve various aspects of services provided to large population dependent on agriculture. Problem of information extraction for domains such as agriculture is particularly challenging due to non-availability of any tagged corpus.

Several domain-specific *named entities (NE)* occur in the documents (such as news) related to the agriculture domain- CROP: names of the crops including varieties; DISEASE: names of the crop diseases and disease causing agents such as pathogen (bacteria, viruses, fungi), insects etc.; CHEMICAL\_TREATMENT: names of pesticides, insecticides, fungicides etc. used in the treatment of a crop disease. Consider an example of NEs tagged for agricultural information extraction:

We usually spray [soybeans]<sub>CROP</sub> with a [strobilurin fungicide]<sub>CHEM\_TREATMENT</sub> because of the potential for [soybean rust]<sub>DISEASE</sub> and other diseases.

As there are few, if any, tagged corpora of agriculture-related documents, we consider an unsupervised approach for extracting these NEs. *NE extraction (NEX)* problem consists of automatically constructing a gazette containing example instances for each NE of interest. A *NE recognition (NER)* algorithm basically matches the given gazette  $G$  (for a particular NE, say DISEASE) with the given document  $D$  to identify occurrences of the instances from  $G$  in  $D$ . While gazette-based NER is fast, the accuracy depends on the quality and completeness of the gazette.

In this paper, we present an enhanced version

<sup>2</sup>[http://en.wikipedia.org/wiki/Economy\\_of\\_India](http://en.wikipedia.org/wiki/Economy_of_India) (access date 31-Jan-2013)

and improved results of our bootstrapping approach to NEX (Patil et al., 2013) and establish its domain independence. We demonstrate the use of this NEX technique for creating gazettes of NE in the agriculture and mechanical engineering domains. Apart from the new application domains for NE extraction, other specific contributions of this paper are as follows: We propose a new variant of the well-known information distance (Bennett et al., 1998; Li et al., 2004; Bu et al., 2010) measure, to decide whether a candidate phrase is a valid instance of the NE or not. We use additional tools and show their effectiveness on improving the gazette quality: (i) a candidate generation algorithm based on maximum entropy classifier; and (ii) a post processing algorithm in the spirit of the assessor module in (Etzioni et al., 2005), but we use statistical hypothesis-testing. Utility and effectiveness of our method is evident from its ability to discover new named entities. For instance, using a limited news corpus, we discovered 8 new crop diseases which are not mentioned in Wikipedia.

The rest of the paper is organized as follows. Section 2 contains a brief overview of related work. In Section 3, we summarize information distance and multiword expression distance, along with new extensions. In Section 4, we discuss our unsupervised algorithm for gazette creation. In Section 5, we present experimental results. Finally we conclude with an outline of future work.

## 2 Related Work

Thelen and Riloff (2002) propose a bootstrapping algorithm called BASILISK for NEX. Etzioni et al. (2005) present a system called KNOWITALL, which implements an unsupervised domain-independent, bootstrapping approach to generate large facts of a specified NE (such as CITY or FILM) from the Web. Many other unsupervised approaches have been proposed for both NEX and NER: (Collins and Singer, 1999; Kim et al., 2002; Meulder and Daelemans, 2003; Talukdar et al., 2006; Jimeno et al., 2008; Liao and Veeramachaneni, 2009; Palshikar, 2012). The basic structure of the bootstrapping approach to NEX is well-known: starting with a seed list of examples of a particular NE type, iteratively identify other phrases which are “similar” to them and add them to the seed list. The algorithm terminates either after a specified number of iterations, or on reaching

a specified gazette size, or when no new entries get added or when the new entities show a “drift”.

## 3 Information Distance for NE

Information distance (Bennett et al., 1998) is an abstract and universal domain-independent, statistically motivated distance measure based on the concept of *Kolmogorov complexity*. Given a Universal Turing Machine (UTM)  $U$ , *Kolmogorov complexity*  $K_U(x|y)$  of a binary string  $x$ , given another binary string  $y$ , is the length of the shortest program for  $U$  that computes  $x$  when given  $y$  as input. Bennett et al. (1998) showed that the quantity (which they called the *information distance*)  $D_{max}(x, y) = \max\{K(x|y), K(y|x)\}$  measures the distance between objects (such as binary strings)  $x$  and  $y$ .

Bu et al. (2010) presented a variant of the information distance, which measures the distance between an  $n$ -gram and its semantics. They define the *context*  $\phi(g)$  of an  $n$ -gram  $g$  as the set of all web-pages containing  $g$  while the *semantics*  $\mu(g)$  of  $g$  is the set of all web-pages that contain all words in  $g$  but not necessarily as a contiguous  $n$ -gram. For example, for  $g = \text{Bill Gates}$ ,  $\phi(g)$  consist of all pages containing `Bill Gates` as the bigram while  $\mu(g)$  consists of all web-pages that contain both `Bill` and `Gates` but not necessarily as a bigram. Clearly,  $\phi(g) \subseteq \mu(g)$ . The *Multiword Expression Distance (MED)*  $MED(g)$  measures the distance of  $g$  from an “intended” (non-compositional) semantics:

$$MED(g) = D_{max}(\phi(g), \mu(g))$$

$$MED(g) \approx \log|\mu(g)| - \log|\phi(g)|$$

Bu et al. (2010) demonstrated the use of MED to perform NER. In this paper, we use a variant of MED to perform NEX. Unlike (Bu et al., 2010), we are constrained to use a corpus rather than the entire Web. Let  $\mathbf{D}$  be a given untagged corpus of sentences. Let  $K$  be a given constant indicating the window size (e.g.,  $K = 3$ ). Let  $g$  be a given candidate phrase. The *context* of  $g$  and a given word  $w$ , denoted  $\phi_K(g, w)$ , is the set of all sentences in  $\mathbf{D}$  which contain both  $g$  (as a  $n$ -gram) as well as  $w$  and further,  $w$  occurs within a window of size  $K$  around  $g$  in that sentence. The *semantics* of  $g$  and a given word  $w$ , denoted  $\mu(g, w)$ , is the set of all sentences in  $\mathbf{D}$  which contain both  $g$  (as an  $n$ -gram) and  $w$ , though  $g$  and  $w$  need

not be within a window of size  $K$  in the sentence. Clearly,  $\phi_K(g, w) \subseteq \mu(g, w)$ . Then we define the *distance* between  $g$  (a given candidate phrase) and a given word  $w$  as follows:

$$MED0_{D,K}(g, w) = \log|\mu(g, w)| - \log|\phi_K(g, w)|$$

Let  $W = \{w_1, w_2, \dots, w_m\}$  be a given finite, non-empty set of  $m$  words. The definition of  $MED0$  is extended to use a given set  $W$  (rather than a single word  $w$ ) by taking the average of the  $MED0$  distance between  $g$  and each word in  $W$ :

$$MED_{D,K}(g, W) = \frac{MED0_{D,K}(g, w_1) + \dots + MED0_{D,K}(g, w_m)}{m}$$

We assume that a subroutine  $MED(\mathbf{D}, K, W, g)$  returns the MED distance  $MED_{D,K}(g, W)$ , as defined above.

## 4 NEX Using MED

In NEX task, we are given (i) an untagged corpus  $\mathbf{D}$  of documents; and (ii) a seed list  $L$  containing known examples of a particular NE type  $T$ . The goal is to create a gazette containing other instances of the NE type  $T$  that occur in  $\mathbf{D}$ . We first look at some sub-problems and then give the complete algorithm for NEX in Section 4.4.

### 4.1 Pre-processing step

The first task is to identify all phrases in  $\mathbf{D}$  that are likely to be instances of the NE type  $T$ . In the simplest case, all phrases in  $\mathbf{D}$  that have the same syntactic structure as the instances in  $L$  could be considered as candidates. For example, for DISEASE, one could look for all NPs that may begin with at most one adjective (e.g., `gray mold`), followed by one or more nouns and whose head word is a singular common noun (as in `crown rot` or `tissue blight`). We assume such simple logic is implemented in a subroutine *GenCandidates* (not shown in this paper). However, even with such syntactic restrictions, the number of candidate instances is usually very large. For instance, for agricultural domain corpus described in Section 5, around 350,000 candidates were output by *GenCandidates*. It also contains many phrases which are obviously not the instances of  $T$  (e.g., `moderate field tolerance`).

Algorithm *Prune* (Fig. 1) *pre-processes* the list  $C$  of candidates from *GenCandidates* using a

```

function Prune( $\mathbf{D}, C, n_0, L_1, L_2$ )
 $G_1 := \emptyset; G_2 := \emptyset;$ 
for  $i := 0; i < MaxIter; i++$  do
   $H_1 := \emptyset; H_2 := \emptyset;$ 
  Build maximum entropy classifier  $M$  using
  instances in  $L_1 \cup G_1$  and  $L_2 \cup G_2$  as positive and
  negative examples for class  $T$  respectively;
  foreach phrase  $g \in C$  &&  $g \notin G_1 \cup G_2$  do
     $(c, pc) :=$  predicted class of  $g$  using  $M$  along
    with predicted probability for class  $c$ 
    if  $c == T$  then  $H_1 := H_1 \cup \{(c, pc)\}$ 
    else  $H_2 := H_2 \cup \{(c, pc)\}$  endif
  end foreach
  Retain only top  $n_0$  elements in  $H_1$  and in  $H_2$  in
  descending order of  $pc$  values
   $G_1 = G_1 \cup H_1$  // add only the phrases from  $H_1$  to  $G_1$ 
   $G_2 = G_2 \cup H_2$  // add only the phrases from  $H_2$  to  $G_2$ 
end for
return ( $G_1$ )

```

Figure 1: Algorithm *Prune*

self-trained, iterative maximum entropy (MaxEnt) classifier. Some of the major features used by MaxEnt classifier are: words in the phrase, words in context and their POS tags, next and previous verbs, binary features to capture presence of adjective, capitalization, proper nouns within phrase.

### 4.2 Backdrop of a Gazette

The key problem in unsupervised NEX is to decide, using  $\mathbf{D}$ , whether a candidate phrase  $g$  has the same NE type  $T$  as the examples in  $L$ . For example, given  $L = \{\text{gray mold, crown rot, tissue blight}\}$  as a seed list for  $T = \text{DISEASE}$  and  $g = \text{corn rust}$  as a candidate phrase, we need to decide whether  $g$  is a DISEASE or not. The idea is to use the  $MED_{D,K}$  as defined above to accept only those  $g$  which have “low” distance (“high” similarity) between  $g$  and the backdrop of the gazette  $L$ , which is defined as a set  $W$  of words “characteristic” (or strongly indicative) of  $T$ .

Function *GetBackdrop*( $\mathbf{D}, L, K, m_0$ ) (Fig. 2) computes, using  $\mathbf{D}$ , the set  $W$  for a given gazette  $L$ . Essentially, it computes how many times each word  $w$  occurs in the context window of given size  $K$  around every entry in  $L$ . Then it computes a *relevance score* for each word. The *relevance score* is computed as a product of following factors: the entropy  $H$  of the word; the number  $b$  of entries in  $L$  for which it is a context word; and ratio of the total number  $f_L$  of times the word occurs in the context of all entries in  $L$  to its frequency  $f$  in the entire corpus  $\mathbf{D}$ . Finally, it returns the top  $m_0$  words in terms of the highest relevance score values as the backdrop for



```

function GetBackdrop(D, L, K, m0)
W = ∅ // initially empty
h = ∅ // hash table key=word value=count
foreach word w in D do
  foreach entry ui ∈ L do
    compute the frequency f(w,ui) of how many
      times w occurs in the context window of
      size K for ui
  end foreach
  // f(w) = total no. of occurrences of w in D
  // fL(w) = f(w,u1) + f(w,u2) + ... + f(w,uL) = total no. of
  // occurrences of w in the context of all entries in L
  compute the entropy of w as
  
$$H(w) = - \sum_{i=1}^{|L|} \frac{f(w, u_i)}{f_L(w)} \log \left( \frac{f(w, u_i)}{f_L(w)} \right)$$

  b(w) := no. of entries ui ∈ L for which f(w,ui) > 0
  Define score(w) := fL(w) / f(w) × b(w) × H(w)
end foreach
W := select top m0 words in terms of their scores
return(W)

```

Figure 2: Algorithm *GetBackDrop*

*L*. For example, suppose *L* consists of 6 DISEASE instances. For the word `causes`,  $b(\text{causes}) = 6$ ,  $H(\text{causes}) = 1.51$  and  $f(\text{causes}) = 75$ , leading to  $\text{score}(\text{causes}) = 16.9895$ . For the word `technology`,  $b(\text{technology}) = 2$ ,  $H(\text{technology}) = 0.0646$  and  $f(\text{technology}) = 84$ , leading to  $\text{score}(\text{technology}) = 0.2485$ . Clearly, `causes` is much more relevant as a cue word for DISEASE than `technology`. The score gives more importance to words that appear more frequently as well as in the context of more entries in the given gazette. Higher entropy value indicates that the word is used more uniformly in the context of many entries in *L*. Words with a more skewed usage (lower entropy) may be good indicator words for specific entries rather than for all entries in *L*. Such words are not preferred.

### 4.3 Assessing the Gazette

We propose a *post-processing* step to assess and improve the quality of the candidate gazette created, by identifying (and removing) those entries in the candidate gazette which are very unlikely to be true instances of NE type *T*.

Suppose the candidate gazette includes the two phrases  $g_1 = \text{late blight}$  and  $g_2 = \text{wet weather}$ . Suppose we had also been given a small set *Q* of *cue words* for the NE type *T*; e.g., for DISEASE, *Q* could be {`disease`, `cause`, `symptom`}. For a given phrase *g*, we compute two counts:  $c(g) = \text{count of sentences in } \mathbf{D} \text{ which contain the } n\text{-}$

gram *g* and  $cq(g) = \text{count of sentences in } \mathbf{D} \text{ which contain (in any order) both the } n\text{-gram } g \text{ and at least one word in } Q$ . Clearly,  $cq(g) \leq c(g)$ . Let  $\hat{f}(g) = \frac{cq(g)}{c(g)}$ . Clearly,  $0 \leq \hat{f}(g) \leq 1$ . For example,  $\hat{f}(g_1) = 38/90 = 0.422$  and  $\hat{f}(g_2) = 104/642 = 0.162$ .

Let  $0 < f_0 < 1$  be a fixed value; e.g.,  $f_0 = 0.2$  (we shall shortly discuss how to obtain  $f_0$ ). Essentially,  $\hat{f}(g)$  indicates how well the phrase *g* “co-occurs” with words in *Q*. “Low” values of  $\hat{f}(g)$  (e.g., those below  $f_0$ ) indicate that the number of occurrences of *g* drops drastically when you restrict to only those sentences that contain at least one word in *Q*. Such phrases are unlikely to be true instances of *T*. We perform a statistical hypothesis test (called *proportion test*) that the fraction  $\hat{f}(g)$  is greater than the given constant  $f_0$ . The null hypothesis is  $H_0 : f(g) \geq f_0$ , where  $f(g)$  is the “true” proportion for *g*, since the observed proportion varies depending on **D**. The test statistic is

$$\frac{\hat{f}(g) - f_0}{\sqrt{\frac{f_0(1-f_0)}{c(g)}}}$$

which follows the Standard Normal distribution. Hence the probability (*p*-value) of observing a particular value of the test statistic can be computed using standard tables. The null hypothesis is rejected if *p* is less than the given significance level  $\alpha$  (we use  $\alpha = 0.05$ ).

For  $g_1, g_2$ , the test statistic values are 5.270 and  $-2.407$ ; and the *p*-values are 0.999 and 0.008. Thus  $g_2$  is correctly rejected as an unlikely instance for the NE type DISEASE. Note that this step requires the user to provide the set *Q* of cue words for NE type *T*; we found that generally only a few words ( $\leq 10$ ) are enough. We set  $f_0$  to just below the minimum among the values for the current gazette *L*. We assume that this logic is implemented as a function *Assessor*(**D**, *Q*, *L*), where *L* is the gazette containing phrases to be assessed.

### 4.4 Unsupervised Gazette Creation

Algorithm *CreateGazetteMED* (Fig. 3) coordinates various modules described so far in this section. It starts with an initial seed list *L* of instances of a particular NE type *T*. It first calls the algorithm *GenCandidates* to create a list *C* of candidate phrases for *T* using **D** and prunes *C* using the algorithm *Prune*. Then in each iteration, it calls the algorithm *GetBackdrop* to cre-



```

algorithm CreateGazetteMED
input  $\mathbf{D}$  // set of all sentences from the corpus
input  $L = \{g_1, \dots, g_n\}$  // seed list of NE instances
input  $L_2$  // seed list of entity non-instances
input  $Q$  // set of cue words for NE type  $T$ 
input  $K$  // context window size; default = 3
input  $n_0$  // no. of candidate instances; default 50000
input  $h_0$  // threshold for MED; default = 0.2
input  $m_0$  // no. of backdrop words; default = 150
input  $maxIter$  // maximum no. of iterations; default = 15
output  $L$  // gazette with new entries added
 $C := GenCandidates(\mathbf{D})$ 
 $C := Prune(\mathbf{D}, C, n_0, L, L_2)$ 
for  $i = 1; i < maxIter; i++$  do
   $A := \emptyset$  // initially empty
   $W := GetBackdrop(\mathbf{D}, L, K, m_0)$ 
  foreach candidate phrase  $g \in C$  &&  $g \notin L$  do
    if  $MED(\mathbf{D}, K, W, g) \leq h_0$  then
       $A := A \cup \{g\}$ 
    endif
  end foreach
   $L := L \cup A$  // add entries in  $A$  to  $L$ 
end while
 $L := Assessor(\mathbf{D}, Q, L)$  // remove unlikely entries

```

Figure 3: Algorithm *CreateGazetteMED*

ate the set  $W$  of backdrop words for  $T$  using  $L$ . Then it uses the modified MED to measure the similarity of each candidate phrase  $g \in C$  with  $W$  and adds  $g$  to a temporary set  $A$  only if it has a “high” similarity with  $W$  (above a threshold). A configurable number (default 10) of top candidates from set  $A$  are added to  $L$  in each iteration. At the end of  $maxIter$ , final set of candidates in  $L$  is then pruned using the *Assessor* algorithm. We have enhanced the earlier version (Patil et al., 2013) by using weighted backdrop to compute  $MED_{D,K}(g, W)$ . Note that the algorithm contains four independent ways of assessing whether a sequence of words is an instance of  $T$  or not, as implemented in *GenCandidates*, *Prune*,  $MED_{D,K}$  and *Assessor*.

## 5 Experimental Evaluation

**Experimental Setup:** In addition to the agricultural news corpus described in (Patil et al., 2013), we also evaluated the proposed technique on a corpus of 7500 engine repair records from mechanical engineering domain. Goal is to create a gazette of engine part names from the engine repair records. The agriculture corpus consists of 30533 documents in English containing 999168 sentences and approximately 19 million words. It was collected using crawler4j (Ganjisaffar, 2013) by crawling the agriculture news web-

	Crop	Disease	Chem. Treatment	Engine Part
$MED_{D,K}$ with assessor	352 (0.662)	237 (0.928)	332 (0.886)	88 (0.818)
PMI with assessor	419 (0.625)	315 (0.911)	341 (0.883)	92 (0.793)
$MED_{D,K}$ no assessor	372 (0.637)	267 (0.831)	502 (0.671)	91 (0.802)
PMI no assessor	441 (0.603)	361 (0.801)	512 (0.670)	95 (0.779)
BASILISK	352 (0.278)	237 (0.924)	332 (0.386)	100 (0.67)

Figure 4: Number of entries (& precision) in the final gazette for each NE type. (To use the same baseline for comparing precision of the proposed algorithm and BASILISK, we use the gazette size of BASILISK comparable to that of  $MED_{D,K}$  with Assessor.)

sites (websites, 2012)<sup>3</sup>. A sample of seeds used to bootstrap the NEX for each category are as following - CROP: {wheat, cotton, corn, soybean, banana}; DISEASE: {wilt, leaf spot, rust, weevil}; CHEM.TREATMENT: {di-syston, evito, tilt, headline}; ENGINE.PARTS: {piston, gaskets, bearings, crankshaft, cylinder}.

**Results:** Fig. 4 summarizes the gazette sizes along with precision for NE types from both agriculture and mechanical engineering (IC engine parts) corpora. Detection rate for agriculture NE type are shown in Fig. 5. To calculate the precision, all the gazettes created by all the algorithms were manually and independently verified by at least three different human annotators. From these results, we conclude that the proposed technique performs well in *domain independent* manner. Assessor module improves precision for all NE types for both measures  $MED_{D,K}$  and PMI. Post-processing using assessor has positive impact on gazette quality. In this experiment, we used top 1000 candidates produced by algorithm *Prune*, instead of top 5000 used in the earlier version (Patil et al., 2013). We observe significant improvement in precision for all NE types. It is clear that the gazette quality is dependent on the output of algorithm *Prune*. We are investigating their inter-relationship as part of this on-going work.

A sample entries from gazette created for each NE type  $T$  are as follows: CROP: {rr alfalfa, sugarcane, biofuel crops, winter canola};

<sup>3</sup>Permission awaited from the content-owners for public release of the corpus for research purpose.

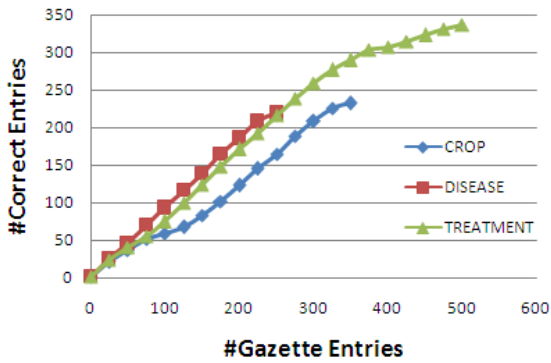


Figure 5: Detection rate of *CreateGazetteMED* with Assessor

DISEASE: { asian soybean rust, alternaria, downy mildew, citrus greening, fusarium wilt}; CHEM.TREATMENT: { ultra blazer, strobilurin, telone II, gaucho grande, spartan}; ENGINE.PARTS:{ piston pin, conn rod, cam gear, cyl head, piston skirt}.

To highlight effectiveness of the gazettes created, we compared our DISEASE gazette with wikipedia. We listed all the page titles on wikipedia falling under categories *Plant pathogens and diseases* (1935) and *Agricultural pest insects* (203). It was quite encouraging to find that, our gazette, though created on a limited size corpus, contained diseases/pathogens not present in wikipedia.<sup>4</sup> Some of these are - limb rot, grape colaspis, black shank, glume blotch, seed corn maggot, mexican rice borer, green bean syndrome, hard lock.

**Comparison with BASILISK:** Our implementation of BASILISK (Pawar et al., 2012) has comparable precision with the proposed *CreateGazetteMED* algorithm for DISEASE category. However, *CreateGazetteMED* clearly outperforms BASILISK for all other categories. Major reason behind worse performance of BASILISK is that it scores each occurrence of the phrase in the corpus independently and the decision to add that phrase to the gazette depends on the highest among all of these scores. *CreateGazetteMED* computes a single score for each phrase by combining evidences from all of its occurrences in the corpus.

**Comparison with KNOWITALL:** KNOWITALL (Etzioni et al., 2005) is a leading, web-scale unsupervised information extraction engine. We implemented basic version of KNOWITALL

algorithm and executed it on above mentioned corpus of agricultural news items. For agriculture domain, KNOWITALL extracted gazettes of following sizes for the three NE types - CROP : 36 entries (20 correct); DISEASE : 55 entries (49 correct); CHEM.TREATMENT : 13 entries (12 correct).

We believe that reason for KNOWITALL's limited gazette size lies in inherent difference between a web-scale search vis-a-vis searching a given corpus. This results in skewed search query statistics and affects the size of gazettes created.

**Comparison with PMI:** To gauge the effectiveness of  $MED_{D,K}$  as a proximity measure, we compare it with PMI (Bouma, 2009). We follow exactly the same steps as described in our *CreateGazetteMED* algorithm with the only difference being use of PMI, instead of  $MED_{D,K}$ . For the results with top 1000 candidates from Prune (Fig. 4),  $MED_{D,K}$  compares favorably with PMI as a proximity measure for all the NE types. Comparing with results with top 5000 candidates from Prune (in (Patil et al., 2013)), we observe that  $MED_{D,K}$  is a more robust proximity measure than PMI. Sensitivity of these measures to output of pre-processing step is part of future work.

## 6 Conclusions and Further Work

We presented improved results of our unsupervised NEX technique, *CreateGazetteMED*. A new variant of MED is used to quantify the proximity (similarity) of a candidate phrase with a given NE type. We established its domain independence using corpora from agriculture and mechanical engineering domains. Effectiveness of *CreateGazetteMED* was validated using experimental comparison with PMI, BASILISK and KNOWITALL. Our method incorporated a pre-processing step (based on MaxEnt classifier). We also proved efficacy of statistical hypothesis testing as a post-processing step to improve gazette quality. As part of further work, developing a robust stopping criterion for automatically stopping the gazette creation process needs attention. Unsupervised relation extraction (such as relations between CROP, DISEASE, CHEMICAL.TREATMENT and many other NEs) is a natural extension. Establishing language independence of the proposed technique, exploring effect of number and quality of initial seeds are also promising avenues.

<sup>4</sup>Verified on 30<sup>th</sup> January, 2013

## References

- C.H. Bennett, P. Gacs, M. Li, P.M.B. Vitanyi, and W.H. Zurek. 1998. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423.
- G. Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*.
- F. Bu, X. Zhu, and M. Li. 2010. Measuring the non-compositionality of multiword expressions. In *Proceedings Of the 23rd Conf. on Computational Linguistics (COLING)*.
- M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP)*.
- O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, and D.S. Weld. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165:91–134.
- Yasser Ganjisaffar. 2013. <http://code.google.com/p/crawler4j/>. [Online; accessed 16 Aug. 2013].
- A. Jimeno, E. Jimenez-Ruiz, V. Lee, S. Gaudan, R. Berlanga, and D. Rebholz-Schuhmann. 2008. Assessment of disease named entity recognition on a corpus of annotated sentences. *BMC Bioinformatics*, 9(Suppl 3)(S3):1–10.
- J.-H. Kim, I.-H. Kang, and K.-S. Choi. 2002. Unsupervised named entity classification models and their ensembles. In *Proceedings of COLING*.
- M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitanyi. 2004. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264.
- W. Liao and S. Veeramachaneni. 2009. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 58–65.
- F.D. Meulder and W. Daelemans. 2003. Memory-based named entity recognition using unannotated data. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL - Volume 4*, pages 208–211.
- G. K. Palshikar. 2012. Techniques for named entity recognition: A survey. In *Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resources*, pages 191–217. IGI Global.
- S. Patil, S. Pawar, G. K. Palshikar, S. Bhat, and R. Srivastava. 2013. Unsupervised gazette creation using information distance. In *Proceedings of the 18th International Conference on Application of Natural Language to Information Systems (NLDB), LNCS 7934*. Springer-Verlag.
- S. Pawar, R. Srivastava, and G. K. Palshikar. 2012. Automatic gazette creation for named entity recognition and application to resume processing. In *Proceedings of ACM COMPUTE*.
- P.P. Talukdar, T. Brants, M. Liberman, and F. Pereira. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 141–148.
- M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Agriculture News Source websites. 2012. [cornandsoybeandigest.com](http://cornandsoybeandigest.com), [deltafarmpress.com](http://deltafarmpress.com), [southwestfarmpress.com](http://southwestfarmpress.com), [southeastfarmpress.com](http://southeastfarmpress.com), [westernfarmpress.com](http://westernfarmpress.com). [Online; accessed Mar. 2012].

# Feature-based Neural Language Model and Chinese Word Segmentation

Mairgup Mansur, Wenzhe Pei, Baobao Chang

Key Laboratory of Computational Linguistics, Ministry of Education

Institute of Computational Linguistics, Peking University

mairgup@gmail.com, williampei1988@126.com, chbb@pku.edu.cn

## Abstract

In this paper we introduce a feature-based neural language model, which is trained to estimate the probability of an element given its previous context features. In this way our feature-based language model can learn representation for more sophisticated features. We introduced the deep neural architecture into the Chinese Word Segmentation task. We got a significant improvement on segmenting performance by sharing the pre-learned representation of character features. The experimental result shows that, while using the same feature sets, our neural segmentation model has a better segmenting performance than CRF-based segmentation model.

## 1 Introduction

Nowadays, as a distributed representation learning framework, Deep Learning Architecture has received a lot of attention in the NLP literature.

As Hinton introduced the idea of distributed representation for symbolic data in (Hinton, 1986), this idea has been a research hot spot for more than twenty years.

Bengio (2003) applied Hinton's idea in the context of language modeling, and developed a neural language model. Mikolov (2011) improved Bengio's neural language model by adding recurrence to the hidden layers, allowing it to beat the state-of-the-art  $n$ -gram model not only in terms of perplexity but also in terms of word error rate in speech recognition. Schwenk (2012) applied similar models in statistical machine translation and improved the BLEU score by almost 2 points.

Different from the traditional  $n$ -gram model, a distributed representation of word can be learned by neural language model from unlabeled raw data. As the most important philosophy of Deep

Learning architecture, the property of learning distributed word representation of neural language model have been proved very useful in NLP tasks.

Colobert et al. (2011) developed the SENNA system that shares word representations across the tasks of language modeling, part-of-speech tagging, chunking, named entity recognition, semantic role labeling and syntactic parsing, which approaches or surpasses the state-of-the-art on these tasks. More interestingly, their experimental result shows that, sharing the word representations that are learned by the neural language model on massive raw data can significantly improve the overall performance of the other tasks.

All these research have showed us the very good capability of the neural language model in learning word representation. However, for many NLP tasks, not only the distributed representation of word, but also the representation of features is important. Since the neural language model can learn the representation of words, can we use it to learn representation of features?

In this paper, we introduce a more generalized feature-based neural language model, which can learn distributed representation of features, which is very useful for many NLP tasks.

Traditional neural language model aims to estimate the probability of a word given words in its previous context, our feature-based neural language model is a generalization of the traditional neural language model, which views the language modeling problem as feature-based prediction problem. The features used to predict an element can be words or characters as well as other sophisticated features extracted from the previous context. After training, our feature-based neural language model can learn a distributed representation of those sophisticated features.

To test the usefulness of the feature representation learned by our feature-based neural language model, we introduce a deep neural archi-

ture into the Chinese Word Segmentation task. We conducted a series of experiments on the SIGHAN-2005 datasets, and got a positive result.

By sharing the feature representation learned by our feature-based neural language model with our neural segmentation model, we got a significant improvement on the segmentation performance.

We also made comparison between our neural segmentation model and the classical segmentation method based on Conditional Random Fields(CRF). The experimental result shows that, while using the same feature sets, the neural segmentation model have a better performance than that of CRF-based method, especially when sharing the pre-learned feature representations.

The rest of the paper is organized as follows. Section 2 overviews the task of Chinese Word Segmentation. Section 3 introduce our feature-based neural language model. The deep neural architecture used for segmentation is described in Section 4. Section 5 gives the experimental result. The last section concludes this paper.

## 2 Chinese Word Segmentation

Unlike English and other western languages, Chinese do not delimit words by white-space. Therefore word segmentation is a very basic and important pre-process for Chinese language processing.

Traditional word segmentation approaches are lexicon-driven (Liang, 1987). Lexicon-driven methods assume that predefined Chinese word lexicon is available, hence the segmentation performance strongly depends on the predefined lexicon.

Xue (2003) proposed a novel way of segmenting Chinese texts, and views the Chinese word segmentation task as a character tagging task. According to Xue’s approach, a tagging model is learned from manually segmented training texts, and then used to assign each character a tag indicating the position of this character within a word it belongs to. Xue’s approach, which did not require any predefined lexicon and have a high performance, became the most popular approach to Chinese word segmentation in recent years.

In Sighan Bakeoff-2005, two participants (Low, 2005) and (Tseng, 2005) have given the best results in almost all word segmentation tracks. Both of their systems use sequence tagging methods based on Conditional Random Field (CRF).

CRF is a statistical sequence modeling framework first introduced into natural language pro-

cessing in (Lafferty et al., 2001). Peng et al. (2004) first used this framework for Chinese word segmentation by treating it as a binary decision task, such that each character is labeled either as the beginning of a word or the continuation of one.

In this paper, we use a CRF-based segmentation system to do a series of comparative experiments.

As in most other work did on segmentation, we use a 4-tag schema, such that each Chinese character is labeled by a tag in the tag set “B,M,E,S”. In which, “B”, “E” and “M” stands for the character in the beginning, ending or middle of a word, “S” means that the character is a word by itself.

We use the following feature templates, which are widely used in most segmentation work:

(a)  $C_n(n = -2, -1, 0, 1, 2)$

(b)  $C_n C_{n+1}(n = -2, -1, 0, 1)$

(c)  $C_{-1} C_1$

Here  $C$  refers to a character;  $n$  refers to the position index relative to the current character. By setting the above feature templates, we actually set a 5-character window to extract features, the current character, 2 characters to its left and 2 to its right.

Other work on segmentation used much more sophisticated feature templates other than the one introduced above. However, defeating the state-of-the-art segmentation work is not the main purpose of this paper. Here, we just want to test whether our model can use the same feature set more efficiently than the CRF-based model. Since it is practicable but not necessary to use other sophisticated feature templates to do the model comparison, we just use the aforementioned feature template as the standard feature set for the Chinese word segmentation task.

## 3 Feature-based Neural Language Model

In this section, we first overview the widely used traditional neural language model. Then we introduce our feature-based neural language model, which is a generalization of the traditional one and can learn representation of features from unlabeled raw data through unsupervised training.

### 3.1 Traditional Neural Language Model

Language models are widely used in many NLP applications to compute “scores” describing how likely a piece of text is. The classical  $n$ -gram language model is a direct application of Markov

models, estimating the probability of a word given its previous words in a sentence.

Neural language models were proposed by Bengio and Ducharme (2003) and Schwenk and Gauvain (2004). These neural language models were designed to estimate the conditional probability of a word given its previous context in a sentence.

Different from the traditional  $n$ -gram language model, neural language model can learn a distributed representation of words from unlabeled raw data. As an application of the deep learning architecture, what we value most is the representation learning ability of the neural language model.

For many NLP tasks, the distributed representation of features is important as well as the one of words. However, the traditional neural language model have this shortcoming that it is designed to learn representation of words only.

To overcome this shortcoming of the traditional neural language model, we propose a much more generalized neural language model that can learn distributed representation of features. To distinguish from the traditional one, we call it the feature-based neural language model.

### 3.2 Feature-based Neural Language Model

Unlike the traditional neural language model, which aims to estimate the probability of a word given its previous context words, our feature-based neural language model views the language modeling problem as feature-based prediction problem. Our feature-based neural language model estimate the probability of an element given its history context feature set. The predicted element could be a word or a character in a sentence.

The architecture of our feature-based neural language model is summarized in Figure 1.

More formally, the probability estimated by our feature-based neural language model is:

$$p(E|F_{history}, \theta)$$

where  $\theta$  is the set of parameters of our model. We denote the feature set extracted from the history context of an element  $E$  as  $F_{history}$ :

$$F_{history} = \{f_1, f_2, \dots, f_k\}$$

The first layer of our feature-based neural language model is called lookup table layer, which maps each  $f \in F_{history}$  into a  $d_f$ -dimensional feature vector  $W_{.f} \in \mathbb{R}^{d_f}$ :

$$LT_W(f) = W_{.f}$$

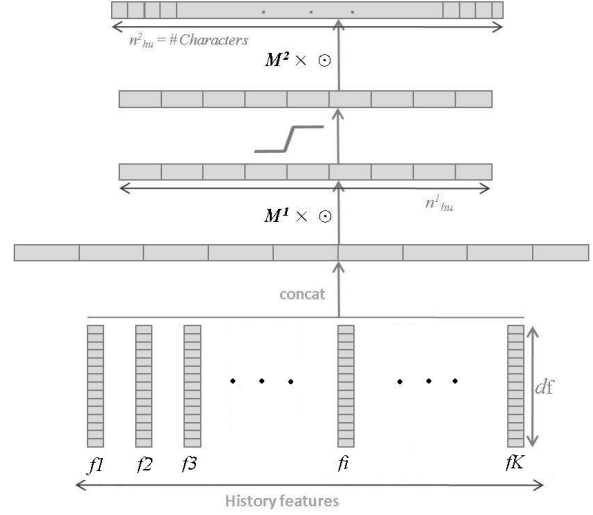


Figure 1: Feature-based Neural Language Model

where  $W$  is called the lookup table, which contains the distributed representation of features, and it is the most important parameter of our feature-based neural language model.

Given a set of features  $F_{history} = [f]_1^k$ , the lookup table layer applies the same operation for each feature in  $F_{history}$ , producing the following  $d_f \times k$  matrix:

$$LT_W([f]_1^k) = (W_{.[f]_1} \quad W_{.[f]_2} \quad \dots \quad W_{.[f]_k})$$

This matrix can be viewed as a  $d_f k$ -dimensional vector  $z^1$  by concatenating its column vectors. Then  $z^1$  can be fed to further neural network layers which perform affine transformations as below:

$$z^l = M^l z^{l-1} + b^l$$

where  $M^l \in \mathbb{R}^{n_{hu}^l \times n_{hu}^{l-1}}$  and  $b^l \in \mathbb{R}^{n_{hu}^l}$  are the parameters to be trained. The hyper-parameter  $n_{hu}^l$  indicates the number of hidden units of the  $l^{th}$  layer. If the  $l^{th}$  layer is not the last layer, to extract highly non-linear features, a non-linearity function must follow. We use  $Tanh()$  to be our non-linearity function.

$$Tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

Finally, the output size of the last layer  $L$  of the network is equal to the number of possible elements. Each output can be then interpreted as the corresponding probability of an element given the history context feature set.

### 3.3 Model Training

All the parameters of our feature-based neural language model are trained by maximizing a likelihood over the unlabeled training data, using stochastic gradient ascent method (Bottou, 1991).

If we denote  $\theta$  to be all the trainable parameters of the network, which are trained using a training set  $\mathcal{T}$  we want to maximize:

$$\theta \mapsto \sum_{(f_h, e) \in \mathcal{T}} \log p(e | f_h, \theta) \quad (1)$$

where  $e$  is an element and  $f_h$  is the corresponding history context feature set of  $e$ .

We maximize equation (1) with stochastic gradient method, by iteratively selecting a random example  $(f_h, e)$  and making a gradient step:

$$\theta \leftarrow \theta + \lambda \frac{\partial \log p(e | \mathbf{f}_h, \theta)}{\partial \theta}$$

where  $\lambda$  is a chosen learning rate, which is also one of the hyper-parameters of our model.

Since our feature-based neural language model is trained on the unlabeled raw data, so its training process is an unsupervised learning procedure.

## 4 Deep Neural Architecture For Chinese Word Segmentation

In this section, we introduce the deep neural architecture into the Chinese word segmentation task.

### 4.1 Neural Network Segmentation Model

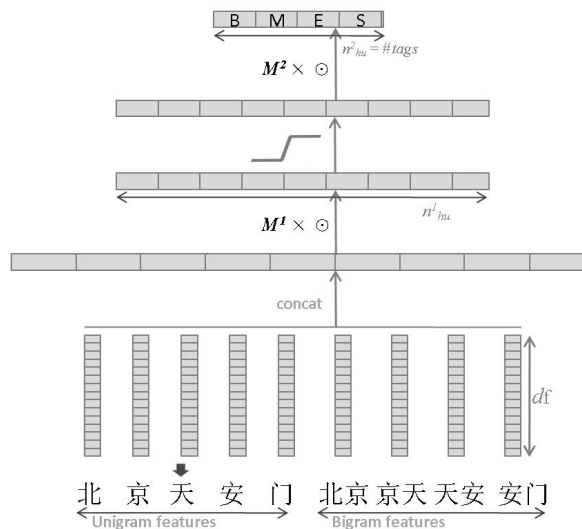


Figure 2: Neural Network Architecture for CWS

The architecture of our neural segmentation model is summarized in Figure 2. We can see that

this architecture is almost the same as our feature-based neural language model, only have slightly difference at the output layer.

As input, character features (unigram or bigram features) are fed as indices taken from a finite dictionary  $\mathcal{D}$ . This dictionary  $\mathcal{D}$  contains all the character features which appeared in the training data.

The first layer of the network maps each of these character feature  $c_f \in \mathcal{D}$  into a  $d_f$ -dimensional feature vector  $W_{\cdot c} \in \mathbb{R}^{d_f}$ , where  $W$  contains distributed representations of the character features. Given a set of character features  $[c_f]_1^T$  in  $\mathcal{D}$ , the lookup table layer applies the same operation for each character feature in the sequence, producing a  $d_f \times T$  matrix.

Given a character to tag, we consider a fixed-size  $k_{sz}$  window of characters around this character to collect the character features. If we only use the unigram features, then there will be  $k_{sz}$  ( $T = k_{sz}$ ) character features that are first passed through the lookup table layer, producing a matrix of fixed size  $d_f \times k_{sz}$ . If we use both the unigram and bigram features, then there will be  $2k_{sz} - 1$  ( $T = k_{sz} + k_{sz} - 1$ ) character features that will produce a matrix in shape of  $d_f \times (2k_{sz} - 1)$ .

And this matrix can be viewed as a  $d_f T$ -dimensional vector by concatenating its column vectors. Then this concatenated vector  $z^1$  can be fed to further neural network layers which perform affine transformations over their inputs.

To extract highly non-linear features at hidden layers, we also use  $Tanh()$  to be our non-linearity function here.

Finally, the output size of the last layer  $L$  of the segmentation model is equal to the number of possible tags, which is 4 (B,M,E,S) in our case. Each output can be then interpreted as probability of tagging the current Chinese character with the corresponding tag.

### 4.2 Model training

All the parameters of our neural segmentation model are trained by maximizing a likelihood over the training data, using stochastic gradient ascent, like we did in previous section.

Since the lookup table of feature vectors is the most important parameter in this neural segmentation model, sharing a well trained lookup table must have a positive impact on the segmentation performance.

We will use our feature-based neural language

model to pre-train a lookup table on a larger raw text data, and then use this pre-trained lookup table to initialize the corresponding lookup table parameter of the neural segmentation model.

Since the pre-training process is unsupervised, so the whole training process of our segmentation model is actually a semi-supervised procedure.

## 5 Experiments

### 5.1 Experimental Setup

To make a comprehensive comparison between our neural segmentation model and the CRF-based model, we conduct a closed test on the PKU dataset from the Sighan-2005 Chinese word segmentation bake-off competition.

We remove the the white-spaces between words in the training and testing data sets of PKU and MSRA datasets and make it all a raw text training data for our feature-based neural language model. Some statistical information of the data sets are given in Table 1.

	PKU	MSRA
Word Types	55,303	88,119
Words	1,109,947	2,368,391
Character Types	4,698	5,167
Characters	1,826,448	4,050,469

Table 1: Statistics of Sighan-2005 data sets.

For evaluation, we use the standard bake-off scoring program to calculate precision, recall, F1, OOV and IV word recall.

To make a performance comparison between our neural segmentation model and the CRF-based model under the same feature sets, we did experiments under three different scenarios.

In each scenario, both our neural segmentation model and CRF-based model are trained using the same feature sets described in Table 2.

As shown in Table 2, in scenario 1, both two models use unigram character features only, in scenario 2 the unigram and bigram character features are used, and in scenario 3 the standard feature set for the CRF-based model.

In each scenario, a feature-based neural language model is trained to learn the distributed representation (lookup table) of the features that is used by the neural segmentation model.

For each experimental setup, we provide the results of our neural segmentation model with

	Feature Sets
Scenario 1	$C_{-2}, C_{-1}, C_0, C_1, C_2$
Scenario 2	$C_{-2}, C_{-1}, C_0, C_1, C_2$ $C_{-2}C_{-1}, C_{-1}C_0, C_0C_1, C_1C_2$
Scenario 3	$C_{-2}, C_{-1}, C_0, C_1, C_2$ $C_{-2}C_{-1}, C_{-1}C_0, C_0C_1, C_1C_2$ $C_{-1}C_1$

Table 2: Feature Sets used in 3 scenarios.

or without sharing the look up table parameters learned by our feature-based language model.

In this paper, we use an open source toolkit “CRF++”<sup>1</sup> to train the CRF models. While training the CRF models, we set the parameter cutoff threshold for the features to be 3 and other parameters are set by default. The hyper-parameters of our feature-based neural language model and neural segmentation model are set as in Table 3.

Window size	$k_{sz} = 5$
Feature dim	$d_f = 30$
Hidden units	$n_{hu} = 300$
learning rate	$\lambda = 0.001$

Table 3: Hyper-Parameters of our neural segmentation model and feature-based neural language model.

### 5.2 Scenario 1: Only Unigram Features

In Table 4 are given the results of experiments on scenario 1, in which only unigram character features are used by both segmentation models.

As the result shows, while using the same feature set, our neural segmentation model have a better performance than that of the CRF-based model, which means our model makes use of the features more efficiently than the CRF-based model.

From the result we can see that sharing the lookup table parameters with the pre-trained feature-based neural language model can significantly improve the performance of our neural segmentation model. The OOV recall is boosted from 48.9% to 68.8%, this means pre-trained lookup table can help neural segmentation model to deal with OOV problem more smoothly.

### 5.3 Scenario 2: Unigram & Bigram Features

The results in Table 5 shows the performance of our neural segmentation model and the CRF-

<sup>1</sup>crfpp.sourceforge.net



	P	R	F1	$R_{OOV}$	$R_{IV}$
CRF	0.846	0.863	0.855	0.533	0.866
NN	0.871	0.879	0.875	0.489	0.895
NN+LM	0.912	0.927	0.920	0.688	0.926

Table 4: Results of our Neural Segmentation Model (NN) and CRF-based Segmentation Model, using only unigram features, “+LM” means sharing pre-trained lookup table parameters.

based model when unigram and bigram features are used. The experimental result again shows that the performance of our neural segmentation model surpasses that of the CRF-based model, while using the same feature set.

We can see that using our feature-based neural language model to pre-train the lookup table parameters can significantly boost the system performance, which means our feature-based neural language model can learn a better lookup table that contains task-useful information.

	P	R	F1	$R_{OOV}$	$R_{IV}$
CRF	0.918	0.934	0.926	0.566	0.940
NN	0.927	0.933	0.930	0.566	0.949
NN+LM	0.932	0.940	0.936	0.628	0.950

Table 5: Results of our Neural Segmentation Model (NN) and CRF-based Segmentation Model, using unigram and bigram features, “+LM” means sharing pre-trained lookup table parameters.

#### 5.4 Scenario 3: All Standard Features

The results in Table 6 shows the performance of our neural segmentation model and the CRF-based model when the standard feature set are used.

The  $+LM_1$  in Table 6 means that neural segmentation model only share unigram feature representations, which are learned by our feature-based language model in scenario 1. The  $+LM_2$  means neural segmentation model share the representations of the standard feature set learned by our feature-based language model.

We can see that our neural segmentation model have almost the same but still better performance than that of CRF-based model. The result also shows that, only sharing pre-learned unigram feature representation is not helpful to boost the performance of our neural segmentation model. When sharing the pre-learned representations of the same features used by neural segmentation model, a significant improvement on the system performance can be achieved.

From Table 4, 5 and 6 we can see that, sharing pre-learned feature representations can always boost the performance of our neural segmentation model. This means that our feature-based neural language model is effective in learning feature representations, even if the features are more sophisticated. The performance boost mainly come from the improvement on OOV recall, means that the pre-learned feature representations is helpful while dealing with the OOV problem in Chinese Word Segmentation task.

	P	R	F1	$R_{OOV}$	$R_{IV}$
CRF	0.924	0.939	0.931	0.609	0.943
NN	0.936	0.928	0.932	0.579	0.958
$NN+LM_1$	0.935	0.929	0.932	0.569	0.958
$NN+LM_2$	0.940	0.939	0.940	0.695	0.955

Table 6: Results of our Neural Segmentation Model (NN) and CRF-based Segmentation Model using standard features, “+LM” means sharing pre-trained lookup table parameters.

## 6 Conclusion

In this paper we introduced a generalized feature-based neural language model, which is trained to estimate the probability of an element given its previous context features, thus making it possible to learn distributed representation of more sophisticated features, which is useful for NLP tasks.

To test the efficiency of the feature representation learned by our feature-based neural language model, we introduced the deep neural architecture into the Chinese Word Segmentation task. We conducted a series of experiments on the Sighan-2005 PKU-dataset.

Experimental result shows that our neural segmentation model have a better performance than that of CRF-based model, while these two models are using the same feature sets.

By sharing the representation learned by our feature-based neural language model with the neural segmentation model, we got a significant improvement on system performance. This proved that useful feature representation can be learned by our feature-based neural language model.

## Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant No. 61273318.

## References

- Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.
- Léon Bottou. 1991. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes*, 91:8.
- 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.
- Geoffrey E Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, pages 1–12. Amherst, MA.
- J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conf. on Machine Learning*, pages 282–289.
- Nanyuan Liang. 1987. ”written chinese word segmentation system—cdws”. *Journal of Chinese Information Processing*, 21:9–19.(in Chinese).
- Jin Kiat Low. 2005. A maximum entropy approach to chinese word segmentation. *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea*, pages 161–164.
- Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.
- Fuchun Peng, Fangfang Feng, and McCallum Andrew. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conf. on Computational Linguistics*, pages 562–568.
- Holger Schwenk and Jean-Luc Gauvain. 2004. Neural network language models for conversational speech recognition. In *ICSLP 2004*.
- Holger Schwenk, Anthony Rousseau, and Mohammed Attik. 2012. Large, pruned or continuous space language models on a gpu for statistical machine translation. *NAACL-HLT 2012*, page 11.
- Huihsin Tseng. 2005. A conditional random field word segmenter for sighan 2005. *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea*, pages 168–171.
- Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8:29–48.

# Human-Computer Interactive Chinese Word Segmentation: An Adaptive Dirichlet Process Mixture Model Approach

Tongfei Chen<sup>1</sup>, Xiaojun Zou<sup>2</sup>, Weimeng Zhu<sup>1</sup>, Junfeng Hu<sup>2,\*</sup>

<sup>1</sup> School of Electronics Engineering and Computer Science,  
Peking University, Beijing, 100871, P. R. China

<sup>2</sup> Key Laboratory of Computational Linguistics (Peking University),  
Ministry of Education, Beijing, 100871, P. R. China  
{ctf, zouxj, zwm, hujf}@pku.edu.cn

## Abstract

Previous research shows that Kalman filter based human-computer interactive Chinese word segmentation achieves an encouraging effect in reducing user interventions, but suffers from the drawback of incompetence in distinguishing segmentation ambiguities. This paper proposes a novel approach to handle this problem by using an adaptive Dirichlet process mixture model. By adjusting the hyperparameters of the model, ideal classifiers can be generated to conform to the interventions provided by the users. Experiments reveal that our approach achieves a notable improvement in handling segmentation ambiguities. With knowledge learnt from users, our model outperforms the baseline Kalman filter model by about 0.5% in segmenting homogeneous texts.

## 1 Introduction

As Chinese text is written without natural delimiters such as whitespaces, word segmentation is often the essential first step in Chinese language processing (Liang, 1987). Over the past two decades, various methods have been developed to address this issue (Nie et al., 1994; Sun et al., 1998; Luo et al., 2002; Zhang et al., 2003; Peng et al., 2004; Goldwater et al., 2006). Generally, supervised statistical learning methods are more adaptive and robust in processing unrestricted texts than the traditional dictionary-based methods.

However, in some domain-specific applications, for example ancient Chinese text processing, there is neither enough homogeneous corpora for training a reliable statistical model, nor a well-defined dictionary. In these tasks, unsupervised word segmentation is preferred to utilize the linguistic knowledge derived from the raw corpus itself. Many researches also enable users to take part in the segmentation process, adding expert knowledge to the system (Wang et al., 2002; Li and Chen, 2007). This is quite reasonable since the criteria of word segmentation are dependent on a user or the destination of use in many applications (Sproat et al., 1996).

Zhu et al. (2013) proposed a Kalman filter based human-computer interactive learning model for segmenting Chinese texts depending upon neither lexicon nor any annotated corpus. This approach enables experts to observe and intervene with the segmentation results, while the segmenter learns and adapts to these knowledge iteratively. At the end of this procedure, a segmentation result that fully matches the demand of the user is returned. However, in some complicated cases where segmentation ambiguities exist, the Kalman filter will not converge and keep swapping in two or more states.

To overcome this drawback, we established an adaptive Dirichlet process mixture model (ADPMM) for human-computer interactive word segmentation. ADPMM gradually adapts itself to the knowledge supplied by users through the process of human-computer interaction, notably reducing human interventions by classifying each occurrence of a bigram into its corresponding class. Each generated class bears a tag *separated* or *combined* derived from user interventions; bigrams classified to a class later is judged as *separated* or *combined* according to the class tag. Knowledge learnt from the user can further

---

\* To whom all correspondence should be addressed.

be used to aid the segmentation of homogeneous corpus.

The rest of this paper is organized as follows. The next section reviews related work. The details of our model are elaborated in Section 3. In Section 4, experiments are presented to illustrate the performance of our model. The final section concludes the proposed model and discusses possible future work.

## 2 Related Work

Unsupervised word segmentation is generally based on some predefined criteria, for example *mutual information* ( $mi$ ), to recognize a substring as a word. Sproat and Shih (1990) studied comprehensively in this direction using mutual information. Many successive researches applied different ensemble methods to mutual information (Chien, 1997; Yamamoto and Kenneth, 2001). Sun et al. (2004) designed an algorithm based on the linear combination of  $mi$  and *difference of t-score* ( $dts$ ). Other criteria like *description length gain* (Kit and Wilks, 1999), *assessor variety* (Feng et al., 2004) and *branch entropy* (Jin and Tanaka-Ishii, 2006) were also explored.

Any automatic segmentation has limitations in some way and is far from fully matching the particular need of users. Thus, human-computer interactive strategies are explored to allow users to pass their linguistic knowledge to the segmenter by directly intervening the segmentation process. Wang et al. (2002) developed a sentence-based human-computer interaction inductive learning method. Feng et al. (2006) proposed a certainty-based active learning segmentation algorithm to train an  $n$ -gram language model in an unsupervised learning framework. Li and Chen (2007) further explored a candidate word based human-computer interactive segmentation strategy.

Kalman filters (Kalman, 1960) are based on linear dynamic systems discretized in the time domain. Given parameters, Kalman filters estimates the unobserved state. Zhu et al. (2013) applied Kalman filter model to learn and estimate user intentions in their human-computer interactive word segmentation framework.

A Dirichlet process is a stochastic process that is a distribution whose domain is itself a distribution (Ferguson, 1973). It can also be viewed as an infinite-dimensional generalization of the Dirichlet distribution. It can be used to construct a mixture model with an unknown number of components (West et al., 1993). Dirichlet processes have been used to handle Chinese word

segmentation. Goldwater et al. (2006) explored a bigram model built upon a Dirichlet process to discover contextual dependencies.

## 3 Model

### 3.1 Baseline Model

Sun et al. (1998) proposed *difference of t-score* ( $dts$ ) as a useful complement to *mutual information* ( $mi$ ). They further designed a compound statistical measurement based on the linear combination of  $mi$  and  $dts$ , named  $md$  (Sun et al., 2004). Given any bigram  $xy$ , in terms of  $md(x,y)$  and a threshold  $\Theta$ , whether the bigram should be combined or separated can be determined—when  $md(x,y)$  is greater than  $\Theta$ , the bigram  $xy$  has more chance to be in a word. This model is a reference to our basic model before the human-computer interaction process. The formulae for calculating  $md$  are as follows:

$$mi^*(x, y) = \frac{mi(x, y) - \mu_{mi}}{\sigma_{mi}},$$

$$dts^*(x, y) = \frac{dts(x, y) - \mu_{dts}}{\sigma_{dts}}, \quad (1)$$

$$md(x, y) = mi^*(x, y) + \lambda \times dts^*(x, y),$$

where  $\mu_{mi}$  and  $\mu_{dts}$  are means of  $mi$  and  $dts$  in the corpus;  $\sigma_{mi}$  and  $\sigma_{dts}$  are standard deviations.  $mi^*$  and  $dts^*$  are normalized versions of measure  $mi$  and  $dts$ ;  $\lambda$  is an empirical value.

Meanwhile, there is an optimization where local maxima and minima of  $md$  appear (Sun et al., 2004). Consider a character string  $abcd$ . If  $md(b,c) > md(a,b)$  and  $md(b,c) > md(c,d)$ , then  $bc$  is considered a *local maximum*. *Local minimum* follows a similar definition. Obviously, local maxima are more likely to form words, while local minima are more likely to be separated. To reflect this kind of tendency, we increase the  $md$  values at local maxima by a constant  $s$ , and decrease the  $md$  values at local minima by  $s$ .

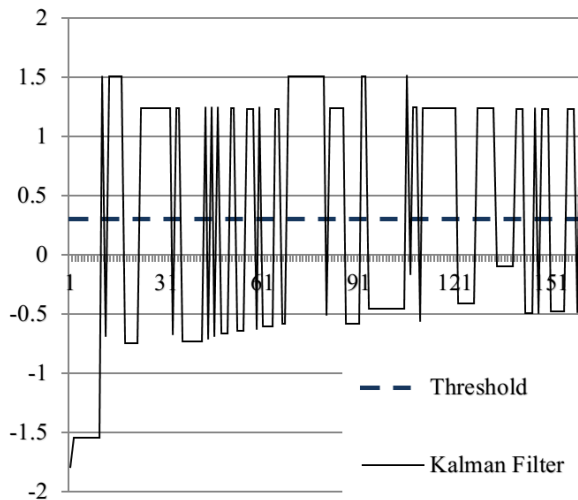
Based on the compound statistical measure  $md$ , Zhu et al. (2013) further developed a human-computer interactive word segmentation framework. In their model, the human interaction process is mapped to a time series process, and user judgments are treated as measurements of the true  $md$  value of bigrams. Each bigram is modeled by a Kalman filter independently to learn and estimate user intentions from user interventions (which may contain noise). Linguistic knowledge is gradually accumulated from the interactions, and eventually, a segmentation that fully matches the specific use is returned.

Both baseline models above use a threshold value  $\Theta$  to classify each bigram into two classes, namely *combined* and *separated*. Both approaches are inherently binary classifiers which seek to classify occurrences of bigrams into classes.

### 3.2 Problem of Segmentation Ambiguity

In the scenario of human-computer interactive Chinese word segmentation, the Kalman filter approach proposed by Zhu et al. (2013) encounters the problem of segmentation ambiguity, rendering it unsuccessful in converging in some special cases. If segmentation ambiguity exists, human interventions would be swapping, which in turn results in swapping states of the Kalman filter.

Take the bigram 及其 used in Zhu et al. (2013) as an example. It exhibits at least two types of segmentation in the corpus (e.g., *separated* in 以及/其他 ‘and others’, and *combined* in 及其/浮动 ‘and its fluctuations’). The Kalman filter approach will not converge, and will keep swapping between two or more states as shown in Figure 1.



**Figure 1.** Problem encountered in Kalman Filter model on the bigram 及其. The vertical axis denotes the *md* value, and the horizontal axis denotes the occurrence of 及其 in the text. An increase in the value denotes that there exist interventions tagged by the user as *combined*; whereas a decrease in the value indicates the presence of interventions tagged as *separated*.

To address this problem, we adopt the *md* measure described by Sun et al. (2004) to construct a Dirichlet process mixture model to classify each occurrence of a bigram into its corresponding class. Each class bears a tag *separated*

or *combined* derived from user interventions. This model gradually adapts itself to the knowledge supplied by the user’s interventions through human-computer interaction, making it more robust in distinguishing segmentation ambiguities through the process of classifying them into different classes.

### 3.3 Adaptive Dirichlet Process Mixture Model

To address the problem mentioned above, classification of each occurrence of a bigram into its corresponding class is required. Since we cannot predict the exact number of classes, a Dirichlet process mixture model (West et al., 1993) would suffice. Similar to the Kalman filter based approach, we also assume that each bigram is independent, i.e., if the model for one bigram changes, other bigrams is not affected. To simplify our discussion, we focus on only one bigram in this section. Notations used in this paper are listed in Table 1.

Symbol	Definition
$\Theta$	Threshold <i>md</i> value
$x_i$	The <i>md</i> value of the <i>i</i> th occurrence of the specific bigram
$\mu_k$	The expectation of the <i>k</i> th class
$\sigma_k^2$	The variance of the <i>k</i> th class
$z_i$	The class indicator of sample $x_i$ , i.e., $x_i$ belongs to class $z_i$
$\alpha$	Concentration parameter of the Dirichlet process mixture model
$H$	Prior base distribution of the Dirichlet process mixture model
$N(x   \mu, \sigma^2)$	Probability density of $N(\mu, \sigma^2)$ at $x$
$H(\mu, \sigma^2)$	Probability density of $H$ at $(\mu, \sigma^2)$
$N\text{-}\Gamma^{-1}$	Normal-inverse-gamma distribution
$\psi$	Prior sum of squared deviations of the mixture model

**Table 1.** Notations used in this paper.

We consider the *md* value of each occurrence of a bigram as a *sample* of the bigram. Initially, samples are classified into class *separated* or *combined* according to threshold value  $\Theta$ . During the interaction process, more classes should be

generated to handle complex situations when binary classifiers are unable to produce correct segmentation result. As the exact number of classes cannot be predicted, the model used for the generation of multiple classes can be formulated as an infinite Gaussian mixture model, in which each sample belongs to a class that follows a Gaussian distribution, and each distribution is specified by a mean and a variance.

Infinite Gaussian mixture models can be formulated by a Dirichlet process with concentration parameter  $\alpha$  and base distribution  $H$  (West et al., 1993):

$$\begin{aligned} G &\sim \text{DP}(\alpha, H), \\ (\mu_k, \sigma_k^2) &\sim G, \\ x_{i, \dots, N} &\sim N(\mu_k, \sigma_k^2). \end{aligned} \quad (2)$$

For simplicity, we choose the prior base distribution  $H$  to be the conjugate prior of  $N(\mu, \sigma^2)$ . The conjugate prior of a Gaussian distribution with unknown expectation and variance is the normal-inverse-gamma distribution:

$$(\mu, \sigma^2) \sim H = \text{N-}\Gamma^{-1}(\mu_0, \kappa, \nu, \psi), \quad (3)$$

where  $\mu_0$  is the prior expectation of  $\mu$  estimated from  $\kappa$  observations, and  $\psi$  is the prior sum of squared deviations estimated from  $\nu$  observations (O’Hagan et al., 2004).

The prior parameter  $\alpha$  and  $\psi$  are of special interest here. Parameter  $\alpha$  is the concentration of the Dirichlet process. The greater  $\alpha$  is, the probability of producing more classes increases. Parameter  $\psi$  represents the prior sum of squared deviations of each class. The lesser  $\psi$  is, the higher precision a class is, and the range the class covers becomes smaller.

To produce more classes that covers smaller ranges, we increase  $\alpha$  and decrease  $\psi$ . We define this step as ADJUSTPARAMETER, which is implemented by multiplying a constant value to  $\alpha$  and  $\psi$  respectively.

In the scenario of human-computer interactive word segmentation, humans can judge whether the segmentation result produced by the segmenter is correct or not. These judgments act as constraints over samples. Classes produced shall conform to these judgments, i.e., samples within each class are uniformly judged as *separated* or uniformly judged as *combined*. If the initial result does not conform to human judgments, more classes with smaller ranges should be generated. Thus ADJUSTPARAMETER should be performed.

Our adaptive Dirichlet process mixture model works as follows: In the initial state of a bigram, we construct a classifier such that all samples

below the threshold  $\Theta$  are marked *separated*, while all samples above  $\Theta$  are marked *combined*. Whenever a user intervention occurred, implying that the current classifier cannot distinguish certain segmentation ambiguities, we increase the concentration parameter  $\alpha$ , i.e. increase the probability to generate more classes, and decrease the prior class sum of squared deviations  $\psi$ , i.e. increase the precision of a class. With these parameters adjusted, re-cluster all the samples to date. Since the prior parameters  $\alpha$  and  $\psi$  are adjusted, the segmenter tends to produce more classes. Iterate this process until all classes conform to human judgments, i.e., samples within each produced class share the same human judgment. Then tag each class with *separated* or *combined* according to the human judgments of the samples in that class. In this way, the Dirichlet process mixture model will adapt itself to conform to human judgments upon samples. This algorithm is illustrated in Algorithm 1.

In Algorithm 1, Line 5 and 6 implements ADJUSTPARAMETER. Empirical values 2.0 and 0.9 are assigned to coefficients  $p_\alpha$  and  $p_\psi$ . The algorithm CLUSTERBYDPM will be elaborated in Section 3.4.

---

#### Algorithm 1. ADAPTIVEDPMM

---

**Input:** Sample set  $X$ , human judgments  $J$

**Output:** Clustering result  $C$

```

1: begin
2:   do
3:      $C \leftarrow \text{CLUSTERBYDPM}(X, \alpha, \psi)$ 
4:     if  $C$  conforms to human judgments  $J$  break
5:      $\alpha \leftarrow \alpha \times p_\alpha$ 
6:      $\psi \leftarrow \psi \times p_\psi$ 
7:   while maximum iteration count not reached
8:   Tag each class in  $C$  according to judgments  $J$ 
9: end

```

---

Whenever a user intervention occurred, the algorithm above is run once, and it returns the expectation and variance of each class, along with the class tag. Use the expectation and variance to construct a naïve Bayes classifier from these data, namely

$$z = \arg \max_k P(k)N(x | \mu_k, \sigma_k^2), \quad (4)$$

where  $x$  is a new sample,  $z$  is the class which  $x$  belongs to, and  $\mu_k, \sigma_k^2$  are the expectation and variance of class  $k$ .  $P(k)$  is the class-prior, i.e. the proportion class  $k$  takes in the whole set of samples. Bigram with  $md$  value  $x$  is judged *separated* or *combined* according the tag associated with class  $z$ . This naïve Bayes classifier is used to classify new occurrences of the bigram until the user intervenes again.

### 3.4 Inference of the Dirichlet Process Mixture Model

Each time a user intervenes in the segmentation process, implying that the samples should be re-clustered, we use a Gibbs sampler to perform the clustering task (MacEachern, 1994; Neal, 2000; Rasmussen, 2000). The algorithm below adopts Algorithm 3 described by Neal (2000).

Set up a Markov chain whose state consists of  $\mathbf{z} = (z_1, \dots, z_n)$ , i.e., the class indicator of current samples. Repeatedly sample as follows:

For  $i = 1, \dots, n$ : Draw a new value for  $z_i$  from:

$$P(z_i = z | z_{-i}, x_i) \propto \begin{cases} \frac{n_{-i,z}}{n-1+\alpha} \int N(x_i | \varphi) H_{-i,z}(\varphi) d\varphi & \text{if } z = z_j \text{ for some } j \neq i \\ \frac{\alpha}{n-1+\alpha} \int N(x_i | \varphi) H(\varphi) d\varphi & \text{if } z_i \neq z_j \text{ for all } j \neq i \end{cases}, \quad (5)$$

where  $\varphi$  indicates the parameter pair  $(\mu, \sigma^2)$ ;  $n_{-i,z}$  is the number of samples in class  $z$  except  $x_i$ ; and  $H_{-i,z}$  is the posterior distribution of  $\varphi$  based on the prior  $H$  and all observations  $x_j$  for which  $j \neq i$  and  $z_j = z$ .

Since  $H$  is chosen to be the conjugate prior of Gaussian distribution, i.e. the normal-inverse-gamma distribution mentioned in Section 3.3, the integral term in Equation (5) is analytically feasible, thus the sampling method presented here is feasible.

## 4 Experiments

In this section, we conducted several experiments to evaluate the performance of our segmentation model. Firstly, we analyzed the performance of segmentation ambiguity handling through a case study. Secondly, we verified the improvement in reducing human intervention after introducing our model. Thirdly, we tested the reusability of knowledge learnt from human interaction. The experiments are based on the *People's Daily* corpus from Jan. 1998 to Jun. 1998 provided by the Institute of Computational Linguistics, Peking University.

Several baseline models are used in this section. One is the approach proposed by Sun et al. (2004) (abbreviated as *Sun's Appr.*) mentioned in Section 3.1, and the other is the Kalman Filter based approach proposed by Zhu et al. (2013) (abbreviated as *Zhu's Appr.*). In addition, the memory approach (abbreviated as *Memory Appr.*), a bigram based human interactive model

whose initial segmentation is exactly the same as Sun's Approach but its prediction of the bigram is taken from the latest human intervention (i.e., the latest correct segmentation result judged by human), is also compared in Sections 4.2 and 4.3. Our adaptive Dirichlet process mixture model is abbreviated as ADPMM.

### 4.1 Case Study

In this part, we took the aforementioned bigram 及其 as an example, and examined the exact number of interventions by users during the human-computer interaction process. Models used for comparison are Memory Appr., Zhu's Appr. and ADPMM. The simulation of the segmentation process was performed by using the correct segmentation text as input to the model. We define an *intervention rate* (IR) of a specific bigram to measure the human effort in a corpus. The IR of bigram  $xy$  is defined as

$$\text{IR}[\%] = \frac{\# \text{ of interventions of } xy}{\# \text{ of occurrences of } xy} \times 100\%. \quad (6)$$

Table 2 shows the number of interventions (denoted by NI) and the IR of bigram 及其 under each model with *People's Daily* Jan. 1998 to Mar. 1998 as test text. It can be seen from the table that ADPMM significantly reduced the number of interventions of bigram 及其 under all three corpora. In Feb. 1998, ADPMM reduced the NI from 36 in Zhu's Appr. to 16 (about 55.56% reduction in percentage), while in Mar. 1998, from 36 to 10 (about 72.22% reduction in percentage). This experiment shows that our model greatly reduced the number of interventions in the case of the segmentation-ambiguous word 及其.

Corpus		Memory Appr.	Zhu's Appr.	ADPMM
Jan.	NI	63	43	<b>17</b>
	IR	39.38	26.88	<b>10.63</b>
Feb.	NI	63	36	<b>16</b>
	IR	42.00	24.00	<b>10.67</b>
Mar.	NI	56	36	<b>10</b>
	IR	25.00	16.07	<b>4.46</b>

**Table 2.** Number of interventions (NI) and IR[%] of bigram 及其 under different corpora.

### 4.2 Simulating the Human-Computer Interactive Segmentation Process

In this part, we simulated the human-computer interaction by using the correct segmentation text as input to the model. We adopted the binary

prediction rate (BPR) described by Zhu et al. (2013) to quantify the conformity of the prediction of the model to user intention. BPR is defined as

$$\text{BPR}[\%] = \frac{\text{\#of correct predictions}}{\text{\#of all predictions}} \times 100\% . \quad (7)$$

The result of the experiment is shown in Table 3. It can be seen that our model gained a slightly higher BPR than both Zhu’s Appr. and Memory Appr. (this is because segmentation ambiguities are relatively rare in corpora), which indicates that our model can reduce user interventions more effectively than Zhu’s Appr.

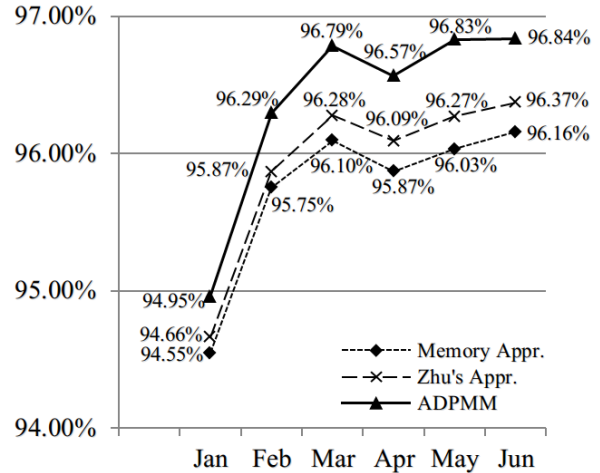
Corpus	Sun’s Appr.	Memory Appr.	Zhu’s Appr.	ADPMM
Jan.	84.22	94.55	94.66	<b>94.95</b>
Feb.	84.58	94.74	94.83	<b>95.14</b>
Mar.	84.59	95.04	95.17	<b>95.46</b>

**Table 3.** BPR[%] of different approaches under different corpora.

### 4.3 Knowledge Reusability Test

After the experiment in Section 4.1, we obtained the classification information for each bigram, and we assumed that this information can be viewed as a kind of learnt knowledge that could be used to aid further word segmentation on homogeneous corpus.

In this part, we performed an incremental test on knowledge reusability. This is done by applying the model with knowledge learnt from text of previous months to segment the text of the current month, from Jan. to Jun., respectively. For example, we took the model with knowledge learnt from Jan. to segment the text of Feb.; the model with knowledge learnt from both Jan. and Feb. to segment text of Mar.; and so on. The BPRs of Memory Appr., Zhu’s Appr. and ADPMM are recorded using the testing scheme described above. As is shown in Figure 2, with the knowledge accumulating, the advantage of our model increases significantly: on Jan. (no previous knowledge exists), the advantage of our model is 0.29% and 0.40% over Zhu’s Appr. and Memory Appr. respectively; on Jun. this advantage is enlarged to 0.47% and 0.68%; on May., this advantage reached 0.56% and 0.80%. This experiments shows that when a large training corpus is present, knowledge of segmentation ambiguities will be stored in our model through the form of different classes of a bigram, making it more robust in handling future segmentation ambiguities.



**Figure 2.** BPR[%] of different word segmentation approaches using an incremental testing scheme.

## 5 Conclusions and Future Work

Research shows that Kalman filter based human-computer interactive Chinese word segmentation framework suffers from the drawback of ineptitude in handling segmentation ambiguities. This paper proposes an adaptive Dirichlet process mixture model (ADPMM). ADPMM adjusts the hyperparameters so that ideal classifiers can be generated to conform to the interventions provided by the users. Experiments showed that our approach achieves a notable improvement (more than 55.56% in a case study) in handling segmentation ambiguities, therefore effective in reducing human effort. In the knowledge reusability test, our model outperforms the baseline Kalman filter model by about 0.5% in segmenting homogeneous texts with knowledge learnt from users.

Our future work will concentrate on improving statistics criteria that would reflect contexts more precisely. As in the experiments, we found that the number of classes may grow rapidly. This is caused by the ineffectiveness of the *md* measure to distinguish different contexts.

### Acknowledgements

We would like to thank Professor Sujian Li for her valuable advice on writing this paper. This work is partially supported by Open Project Program of the National Laboratory of Pattern Recognition (NLPR) and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201102).



## References

- Lee-Feng Chien. 1997. Pat-Tree-Based Keyword Extraction for Chinese Information Retrieval. In *ACM SIGIR Forum*, pages 50-58.
- Michael D. Escobar. 1994. Estimating Normal Means with a Dirichlet Process Prior. *Journal of the American Statistical Association*, 89(425): 268-277.
- Chong Feng, Zhaoxiong Chen, Heyan Huang, and Zhenzhen Guan. 2006. Active Learning in Chinese Word Segmentation Based on Multigram Language Model. *Journal of Chinese Information Processing*, 20(1): 50-58 (in Chinese)
- Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30(1): 75-93.
- Thomas S. Ferguson. 1973. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, pages 209-230.
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual Dependencies in Unsupervised Word Segmentation. In *COLING/ACL 2006*, pages 673-680.
- Zhihui Jin and Kumiko Tanaka-Ishii. 2006. Unsupervised Segmentation of Chinese Text by Use of Branching Entropy. In *COLING/ACL 2006*, pages 428-435.
- Chunyu Kit and Yorick Wilks. 1999. Unsupervised Learning of Word Boundary with Description Length Gain. In *Proceedings of the CoNLL99 ACL Workshop*, pages 1-6.
- Bin Li and Xiaohe Chen. 2007. A Human-Computer Interaction Word Segmentation Method Adapting to Chinese Unknown Texts. *Journal of Chinese Information Processing*, 21(3): 92-98. (in Chinese)
- Rudolph E. Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1): 35-45.
- Nanyuan Liang. 1987. CDWS: An Automatic Word Segmentation System for Written Chinese Texts. *Journal of Chinese Information Processing*, 1(2): 44-52. (in Chinese)
- Xiao Luo, Maosong Sun, and Benjamin K. Tsou. 2002. Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information. In *COLING 2002*, pages 1-7.
- Radford M. Neal. 2000. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2): 249-265.
- Jian-Yun Nie, Wanying Jin, and Marie-Louise Hannan. 1994. A Hybrid Approach to Unknown Word Detection and Segmentation of Chinese. In *Proceedings of International Conference on Chinese Computing*, pages 326-335.
- Anthony O'Hagan, Jonathan Forster, and Maurice G. Kendall. 2004. *Bayesian Inference*. London: Arnold.
- Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese Segmentation and New Word Detection Using Conditional Random Fields. In *COLING 2004*, pages 23-27.
- Carl E. Rasmussen. 2000. The Infinite Gaussian Mixture Model. *Advances in Neural Information Processing Systems*, 12(5.2): 2.
- Richard Sproat, William Gale, Chilin Shih, and Nancy Change. 1996. A Stochastic Finite-State Word Segmentation Algorithm for Chinese. *Computational Linguistics*, 22(3): 377-404.
- Richard Sproat and Chilin Shih. 1990. A Statistical Method for Finding Word Boundaries in Chinese Text. *Computer Processing of Chinese and Oriental Languages*, pages 336-351.
- Maosong Sun, Dayang Shen, and Benjamin K. Tsou. 1998. Chinese Word Segmentation Without Using Lexicon and Hand-Crafted Training Data. In *COLING/ACL 1998*, (1998) 1265-1271.
- Maosong Sun, Ming Xiao, and Benjamin K. Tsou. 2004. Chinese Word Segmentation without Using Dictionary Based on Unsupervised Learning Strategy. *Chinese Journal of Computers*, 27(6): 736-742 (in Chinese)
- Zhongjian Wang, Kenji Araki, and Koji Tochinnai. 2002. A Word Segmentation Method with Dynamic Adapting to Text Using Inductive Learning. In *Proceedings of the First SIGHAN Workshop on Chinese Language Processing*, pages 1-5.
- Mike West, Peter Müller, and Michael D. Escobar. 1993. *Hierarchical Priors and Mixture Models, with Application in Regression and Density Estimation*. Institute of Statistics and Decision Sciences, Duke University.
- Mikio Yamamoto, and Church W. Kenneth. 2001. Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. *Computational Linguistics*, 27(1): 1-30.
- Hua-Ping Zhang, Qun Liu, Xue Q. Cheng, and Hong K Yu. 2003. Chinese Lexical Analysis Using Hierarchical Hidden Markov Model. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 63-70.
- Weimeng Zhu, Ni Sun, Xiaojun Zou, and Junfeng Hu. 2013. The Application of Kalman Filter Based Human-Computer Learning Model to Chinese-Word Segmentation. In *Computational Linguistics and Intelligent Text Processing*, pages 218-230.

# Effect of Non-linear Deep Architecture in Sequence Labeling

**Mengqiu Wang**

Computer Science Department  
Stanford University  
Stanford, CA 94305, USA  
mengqiu@cs.stanford.edu

**Christopher D. Manning**

Computer Science Department  
Stanford University  
Stanford, CA 94305, USA  
manning@cs.stanford.edu

## Abstract

If we compare the widely used Conditional Random Fields (CRF) with newly proposed “deep architecture” sequence models (Collobert et al., 2011), there are two things changing: from linear architecture to non-linear, and from discrete feature representation to distributional. It is unclear, however, what utility non-linearity offers in conventional feature-based models. In this study, we show the close connection between CRF and “sequence model” neural nets, and present an empirical investigation to compare their performance on two sequence labeling tasks – Named Entity Recognition and Syntactic Chunking. Our results suggest that non-linear models are highly effective in low-dimensional distributional spaces. Somewhat surprisingly, we find that a non-linear architecture offers no benefits in a high-dimensional discrete feature space.

## 1 Introduction

Sequence labeling encompasses an important class of NLP problems that aim at annotating natural language texts with various syntactic and semantic information, such as part-of-speech tags and named-entity labels. Output from such systems can facilitate downstream applications such as Question Answering and Relation Extraction. Most methods developed so far for sequence labeling employ generalized linear statistical models, meaning methods that describe the data as a combination of linear basis functions, either directly in the input variables space (e.g., SVM) or through some transformation of the probability distributions (e.g., “log-linear” models).

Recently, Collobert et al. (2011) proposed

“deep architecture” models for sequence labeling (named Sentence-level Likelihood Neural Nets, abbreviated as SLNN henceforth), and showed promising results on a range of tasks (POS tagging, NER, Chunking, and SRL). Two new changes were suggested: extending the model from a linear to non-linear architecture; and replacing discrete feature representations with distributional feature representations in a continuous space. It has generally been argued that non-linearity between layers is vital to the power of neural models (Bengio, 2009). The relative contribution of these changes, however, is unclear, as is the question of whether gains can be made by introducing non-linearity to conventional feature-based models.

In this paper, we illustrate the close relationship between CRF and SLNN models, and conduct an empirical investigation of the effect of nonlinearity with different feature representations. Experiments on Named Entity Recognition (NER) and Syntactic Chunking tasks suggest that non-linear models are highly effective in low-dimensional distributed feature space, but offer no benefits in high-dimensional discrete space. Furthermore, both linear and non-linear models improve when we combine the discrete and continuous feature spaces, but a linear model still outperforms the non-linear one.

## 2 From CRFs To SLNNs

A CRF models the conditional probability of structured output variables  $\mathbf{y}$  given observations  $\mathbf{x}$ . In sequence modeling, the observations are typically words in a sentence, and the output variables are some syntactic or semantic tags we are trying to predict for each word (e.g., POS, named-entity tags, etc.). The most commonly used CRF model has a linear chain structure, where prediction  $y_i$

at position  $i$  is independent of other predictions, given its neighbors  $y_{i-1}$  and  $y_{i+1}$ . It is customary to describe the model as an undirected graphical model, with the following probability definition:

$$P(\mathbf{y}|\mathbf{x}) = \frac{\prod_{i=1}^{|\mathbf{x}|} \Psi(\mathbf{x}, y_i; \Theta) \prod_{j=1}^{|\mathbf{x}|} \Phi(\mathbf{x}, y_j, y_{j-1}; \Lambda)}{Z(\mathbf{x})}$$

$$\Psi(\mathbf{x}, y_i; \Theta) = \exp \left\{ \sum_{k=1}^m \theta_{(k, y_i)} f_k(\mathbf{x}) \right\}$$

$$\Phi(\mathbf{x}, y_i, y_{i-1}; \Lambda) = \exp \left\{ \sum_{k=1}^{m'} \lambda_{(k, y_i, y_{i-1})} g_k(\mathbf{x}) \right\}$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \left( \prod_{i=1}^{|\mathbf{x}|} \Psi(\mathbf{x}, y'_i) \prod_{j=1}^{|\mathbf{x}|} \Phi(\mathbf{x}, y'_j, y'_{j-1}) \right)$$

$\Psi(\mathbf{x}, y_i)$  denotes node clique potentials in this graph, and  $\Phi(\mathbf{x}, y_i, y_{i-1})$  denotes edge clique potentials.  $f_k(\mathbf{x})$  is the set of node-level feature functions,  $m$  is the number of node features, and  $\theta_{(k, y_i)}$  is a weight parameter of feature  $k$  associated with a particular output  $y_i$ ; similarly for edges we have  $g_k(\mathbf{x})$ ,  $m'$ , and  $\lambda_{(k, y_i, y_{i-1})}$ .  $Z(\mathbf{x})$  is the partition function that sums over all possible assignments of output variables in the entire sequence.

Let us focus our discussion on the node clique potentials  $\Psi$  for now. We call the operand of the exponentiation operator in  $\Psi$  a *potential function*  $\psi$ . In a CRF, this can be expressed in matrix notation as:

$$\psi(\mathbf{x}, y_i; \Theta) = |\Theta^\top \mathbf{f}(\mathbf{x})|_{\hat{y}_i 1}$$

We use the notation  $\hat{y}_i$  to denote the ordinal index of the value assigned to  $y_i$ . This linear potential function  $\psi$  can be visualized using a neural network diagram, shown in the left plot in Figure 1. Each edge in the graph represents a parameter weight  $\theta_{(k, \hat{y}_i)}$ , for feature  $f_k(\mathbf{x})$  and a variable assignment of  $y_i$ . In neural network terminology, this architecture is called a single-layer Input-Output Neural Network (IONN).<sup>1</sup> Normalizing locally in a logistic regression is equivalent to adding a *softmax* layer to the output layer of the IONN, which was commonly done in neural networks, such as in Collobert et al. (2011).

We can add a hidden linear layer to this architecture to formulate a two-layer Linear Neural

<sup>1</sup>The bias parameter “ $b$ ” commonly seen in Neural Network convention can be encoded as an “always on” feature in the input layer.

Network (LNN), as shown in the middle diagram of Figure 1. The value of the node  $z_j$  in the hidden layer is computed as  $z_j = \sum_k \omega_{(k, j)} f_k(\mathbf{x})$ . The

value  $y_i$  for nodes in the output layer is computed as:  $y_i = \sum_j \delta_{(j, i)} z_j = \sum_j \delta_{(j, i)} \sum_k \omega_{(k, j)} f_k(\mathbf{x})$ .

where  $\omega_{(k, j)}$  and  $\delta_{(j, i)}$  are new parameters introduced in the model. In matrix form, it can be written as  $\mathbf{y} = \Delta^\top \mathbf{z} = \Delta^\top \Omega^\top \mathbf{f}(\mathbf{x})$ . The node potential function now becomes:

$$\psi'(\mathbf{x}, y_i; \Omega, \Delta) = |\Delta^\top \Omega^\top \mathbf{f}(\mathbf{x})|_{\hat{y}_i 1}$$

This two-layer network is actually no more powerful than the previous model, since we can always compile it down to a single-layer IONN by making  $\Theta = \Omega \Delta$ . In the next step, we take the output of the hidden layer in the LNN, and send it through a non-linear activation function, such as a *sigmoid* or *tanh*, then we arrive at a two-layer Deep Neural Network (DNN) model. Unlike the previous two models, the DNN is non-linear, and thus capable of representing a more complex decision surface.

So far we have extended the potential function used in node cliques of a CRF to a non-linear DNN. And if we keep the potential function for edge cliques the same as before, then in fact we have arrived at an identical model to the SLNN in Collobert et al. (Collobert et al., 2011). The difference between a SLNN and an ordinary DNN model is that we need to take into consideration the influence of edge cliques, and therefore we can no longer normalize the clique factors at each position to calculate the local marginals, as we would do in a logistic regression. The cardinality of the output variable vector  $\mathbf{y}$  grows exponentially with respect to input sequence length. Fortunately, we can use *forward-backward* style dynamic programming to compute the marginal probabilities efficiently.

It is also worth pointing out that this model has in fact been introduced a few times in prior literature. It was termed *Conditional Neural Fields* by Peng et al. (2009), and later *Neural Conditional Random Fields* by Do and Artieres (2010). Unfortunately, the connection to Collobert and Weston (2008) was not recognized in either of these two studies; vice versa, neither of the above were referenced in Collobert et al. (2011). This model also appeared previously in the speech recognition literature in Prabhavalkar and Fosler-Lussier (2010).

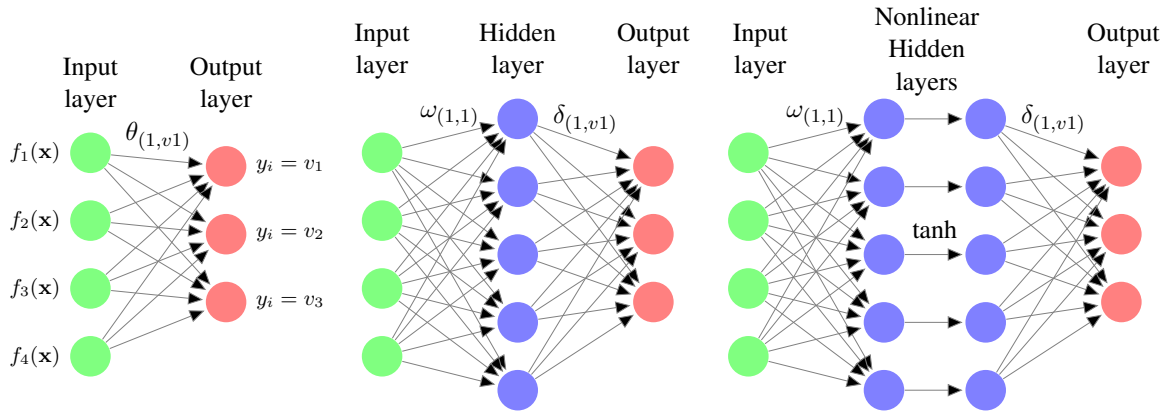


Figure 1: In this diagram, we assume the random variable  $y_i$  has three possible value assignments ( $v_1, v_2, v_3$ ). On the left side is the linear potential function  $\psi$  in CRF, illustrated as a single-layer Input-Output Neural Network. In the middle is a potential function as a two-layer Linear Neural Network; on the right side is a two-layer Deep Neural Network.

### 3 Parameter Learning

Supervised conditional training of the SLNN model amounts to maximizing the objective function  $\mathcal{L}$ , which is given by the sum of log-probabilities over training examples:

$$\mathcal{L}(\mathbf{Y}^*|\mathbf{X}) = \sum_{l=1}^{|\mathbf{X}|} \left( \sum_{i=1}^{|\mathbf{x}^l|} \psi'(\mathbf{x}^l, \mathbf{y}_i^{l*}) + \sum_{j=1}^{|\mathbf{x}^l|} \phi(\mathbf{x}^l, \mathbf{y}_j^{l*}, \mathbf{y}_{j-1}^{l*}) \right) - \sum_{l=1}^{|\mathbf{X}|} \log Z(\mathbf{x})$$

The change in node potential function from  $\psi$  to  $\psi'$  does not affect the inference procedure, and thus we can employ the same dynamic programming algorithm as in a CRF to calculate the log sum over  $Z(\mathbf{x})$  and the expectation of feature parameters.

We adopted the simple L-BFGS algorithm for training weights in this model (Liu and Nocedal, 1989). Although L-BFGS is in general slower than mini-batch SGD – another common optimization algorithm used to train neural networks (Bengio et al., 2006, *inter alia*), it has been found to be quite stable and suitable for learning neural networks (Socher et al., 2011). The gradient of a parameter  $\omega_{(k,j)}$  is calculated as the following:

$$\frac{\partial \mathcal{L}}{\partial \omega_{(k,j)}} = \sum_{l=1}^{|\mathbf{X}|} \sum_{i=1}^{|\mathbf{x}^l|} \left( \frac{\partial \psi'(\mathbf{x}^l, \mathbf{y}_i^l)}{\partial \omega_{(k,j)}} - \mathbb{E}_{P(\mathbf{y}^l|\mathbf{x}^l)} \left[ \frac{\partial \psi'(\mathbf{x}^l, \mathbf{y}_i^l)}{\partial \omega_{(k,j)}} \right] \right)$$

The partial derivative of the potential function  $\frac{\partial \psi'(\mathbf{x}^l, \mathbf{y}_i^l)}{\partial \omega_{(k,j)}}$  can be calculated using the back-propagation procedure, identical to how gradients of a standard Multilayer Perceptron are calculated. The gradient calculation for output layer parameters  $\Delta$  and edge parameters  $\Lambda$  follow the same form. We apply  $\ell_2$ -regularization to prevent overfitting.

### 4 Empirical Evaluation

We evaluate the CRF and SLNN models on two standard sequence labeling tasks: Syntactic Chunking and Named Entity Recognition (NER). In both experiments, we use the publicly available Stanford CRF Toolkit (Finkel et al., 2005).

#### 4.1 Named Entity Recognition

We train all models on the standard CoNLL-2003 shared task benchmark dataset (Sang and Meulder, 2003), which is a collection of documents from Reuters newswire articles, annotated with four entity types: *Person*, *Location*, *Organization*, and *Miscellaneous*. We adopt the BIOES-style annotation standard. Beginning and intermediate positions of an entity are marked with *B*- and *I*- tags, and non-entities with *O* tag. The training set contains 204K words (14K sentences), the development set contains 51K words (3.3K sentences), and the test set contains 46K words (3.5K sentences).

To evaluate out-of-domain performance, we run the models trained on CoNLL-03 training data on two additional test datasets. The first dataset

(ACE) is taken from the ACE Phase 2 (2001-02) and ACE-2003 data. Although the ACE dataset also consists of newswire text and thus is not strictly out-of-domain, there is a genre or dialect difference in that it is drawn from mostly American news sources, whereas CoNLL is mostly English. The test portion of this dataset contains 63K words, and is annotated with 5 original entity types: *Person*, *Location*, *Organization*, *Fact*, and *GPE*. We remove all entities of type *Fact* and *GPE* by relabeling them as *O* during preprocessing, and discard entities tags of type *Miscellaneous* in the output of the models. The second dataset is the MUC7 Formal Run test set, which contains 59K words. It is also missing the *Miscellaneous* entity type, but includes 4 additional entity types that do not occur in CoNLL-2003: *Date*, *Time*, *Money*, and *Percent*. We converted the data to CoNLL-2003 type format using the same method applied to the ACE data.

We used a comprehensive set of features that comes with the standard distribution of Stanford NER model (Finkel et al., 2005). A total number of 437,905 features were generated for the CoNLL-2003 training dataset.

## 4.2 Syntactic Chunking

In Syntactic Chunking, we tag each word with its phrase type. For example, tag *B-NP* indicates a word starts a noun phrase, and *I-PP* marks an intermediate word of a prepositional phrase. We test the models on the standard CoNLL-2000 shared task evaluation set (Sang and Buchholz, 2000). This dataset comes from the Penn Treebank. The training set contains 211K words (8.9K sentences), and the test set contains 47K words (2K sentences). The set of features used for this task is:

- Current word and tag
- Word pairs:  $w_i \wedge w_{i+1}$  for  $i \in \{-1, 0\}$
- Tags:  $(t_i \wedge t_{i+1})$  for  $i \in \{-1, 0\}$ ;  $(t_{-1}, t_0, t_{i+1})$ ;
- The Disjunctive word set of the previous and next 4 positions

A total number of 317794 features were generated on this dataset.

## 4.3 Experimental Setup

In all experiments, we used the development portion of the CoNLL-2003 data to tune the  $\ell_2$ -regularization parameter  $\sigma$  (variance in Gaussian prior), and found 20 to be a stable value. Overall tuning  $\sigma$  does not affect the qualitative results

in our experiments. We terminate L-BFGS training when the average improvement is less than  $1e-3$ . All model parameters are initialized to a random value in  $[-0.1, 0.1]$  in order to break symmetry. We did not explicitly tune the features used in CRF to optimize for performance, since feature engineering is not the focus of this study. However, overall we found that the feature set we used is competitive with CRF results from earlier literature (Turian et al., 2010; Collobert et al., 2011). For models that embed hidden layers, we set the number of hidden nodes to 300.<sup>2</sup> Results are reported on the standard evaluation metrics of entity/chunk precision, recall and F1 measure.

For experiments with continuous space feature representations (a.k.a., word embeddings), we took the word embeddings (130K words, 50 dimensions) used in Collobert et al. (2011), which were trained for 2 months over Wikipedia text.<sup>3</sup> All sequences of numbers are replaced with num (e.g., “PS1” would become “PSnum”), sentence boundaries are padded with token PAD, and unknown words are grouped into UNKNOWN. We attempt to replicate the model described in Collobert et al. (2011) without task-specific fine-tuning, with a few exceptions: 1) we used the *soft tanh* activation function instead of *hard tanh*; 2) we use the BIO2 tagging scheme instead of BIOES; 3) we use L-BFGS optimization algorithm instead of stochastic gradient descent; 4) we did not use Gazetteer features; 5) Collobert et al. (2011) mentioned 5 binary features that look at the capitalization pattern of words to append to the embedding as additional dimensions, but only 4 were described in the paper, which we implemented accordingly.

## 5 Results and Discussion

For both the CRF and SLNN models, we experiment with both the discrete binary valued feature representation used in a regular CRF, and the word embeddings described. Unless otherwise stated, the set of edge features is limited to pairs of predicted labels at the current and previous positions, i.e.,  $(y_i, y_{i-1})$ . The same edge features were used in Collobert et al. (2011) and were called “transition scores” ( $[A]_{i,j}$ ).

<sup>2</sup>We tried varying the number of hidden units in the range from 50 to 500, and the main qualitative results remain the same.

<sup>3</sup>Available at <http://ml.nec-labs.com/senna/>.

	CRF			SLNN		
	P	R	F1	P	R	F1
CoNLL <sub>d</sub>	90.9	90.4	<b>90.7</b>	89.3	89.7	89.5
CoNLL <sub>t</sub>	85.4	84.7	<b>85.0</b>	83.3	83.9	83.6
ACE	81.0	74.2	<b>77.4</b>	80.9	74.0	77.3
MUC	72.5	74.5	<b>73.5</b>	71.1	74.1	72.6
Chunk	93.7	93.5	<b>93.6</b>	93.3	93.3	93.3

Table 1: Results of CRF versus SLNN, over discrete feature space. CoNLL<sub>d</sub> stands for the CoNLL development set, and CoNLL<sub>t</sub> is the test set. Best F1 score on each dataset is highlighted in bold.

## 5.1 Results of Discrete Representation

The first question we address is the following: in the high-dimensional discrete feature space, would the non-linear architecture in SLNN model help it to outperform CRF?

Results from Table 1 suggest that SLNN does not seem to benefit from the non-linear architecture on either the NER or Syntactic Chunking tasks. In particular, on the CoNLL and MUC dataset, SLNN resulted in a 1% performance drop, which is significant for NER. The specific statistical properties of this dataset that lead to the performance drop are hard to determine, but we believe it is partially because the SLNN has a much harder non-convex optimization problem to solve – on this small dataset, the SLNN with 300 hidden units generates a shocking number of 100 million parameters (437905 features times 300 hidden dimensions), due to the high dimensionality of the input feature space.

To further illustrate this point, we also compared the CRF model with its Linear Neural Network (LNN) extension, which has exactly the same number of parameters as the SLNN but does not include the non-linear activation layer. Although this model is identical in representational power to the CRF as we discussed in Section 2, the optimization problem here is no longer convex (Ando and Zhang, 2005). To see why, consider applying a linear scaling transformation to the input layer parameter matrix  $\Omega$ , and apply the inverse scaling to output layer  $\Delta$  matrix. The resulting model has exactly the same function values. We can see from Table 2 that there is indeed a performance drop with the LNN model as well, likely due to difficulty with optimization. By comparing the results of LNN and SLNN, we see that the addition of a non-linear activation layer in SLNN does not seem to help, but in fact further decreases

	CRF			LLN		
	P	R	F1	P	R	F1
CoNLL <sub>d</sub>	90.9	90.4	<b>90.7</b>	89.5	90.6	90.0
CoNLL <sub>t</sub>	85.4	84.7	<b>85.0</b>	83.1	84.7	83.9
ACE	81.0	74.2	<b>77.4</b>	80.7	74.3	77.3
MUC	72.5	74.5	73.5	72.3	75.2	<b>73.7</b>
Chunk	93.7	93.5	<b>93.6</b>	93.1	93.2	93.2

Table 2: Results of CRF versus LNN, over discrete feature space.

performance in all cases except Syntactic Chunking.

A distinct characteristic of NLP data is its high dimensionality. The vocabulary size of a decent sized text corpus is already in the tens of thousands, and bigram statistics are usually an order of magnitude larger. These basic information units are typically very informative, and there is not much structure in them to be explored. Although some studies argue that non-linear neural nets suffer less from the curse of dimensionality (Attali and Pagés, 1997; Bengio and Bengio, 2000; Pitkow, 2012), counter arguments have been offered (Camastra, 2003; Verleysen et al., 2003). The empirical results from our experiment seems to support the latter. Similar results have also been found in other NLP applications such as Text Classification. Joachims concluded in his seminal work: “non-linear SVMs do not provide any advantage for text classification using the standard kernels” (Joachims, 2004, p. 115). If we compare the learning curve of CRF and SLNN (Figure 2), where we vary the amount of binary features available in the model by random sub-sampling, we can further observe that SLNNs enjoy a small performance advantage in lower dimensional space (when less than 30% of features are used), but are quickly outpaced by CRFs in higher dimensional space as more features become available.

Another point of consideration is whether there is actually much non-linearity to be captured in sequence labeling. While in some NLP applications like grammar induction and semantic parsing, the data is complex and rich in statistical structures, the structure of data in sequence labeling is considerably simpler. This contrast is more salient if we compare with data in Computer Vision tasks such as object recognition and image segmentation. The interactions among local variables there are much stronger and more likely to be non-linear. Lastly, models like CRF actually already capture some of the non-linearity in the

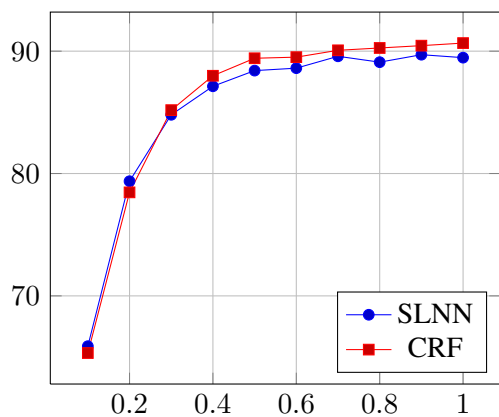


Figure 2: The learning curve of SLNN vs. CRF on CoNLL-03 dev set, with respect to the percentage of discrete features used (i.e., size of input dimension). Y-axis is the F1 score (out of 100), and X-axis is the percentage of features used.

	CRF			SLNN		
	P	R	F1	P	R	F1
CoNLL <sub>d</sub>	80.7	78.7	79.7	86.1	87.1	<b>86.6</b>
CoNLL <sub>t</sub>	76.4	75.5	76.0	79.8	81.7	<b>80.7</b>
ACE	71.5	71.1	71.3	75.8	74.1	<b>75.0</b>
MUC	65.3	74.0	69.4	65.7	76.8	<b>70.8</b>

Table 3: Results of CRF versus SLNN, over continuous space feature representations.

input space through the interactions of latent variables (Liang et al., 2008), and it is unclear how much additional gain we would get by explicitly modeling the non-linearity in local inputs.

## 5.2 Results of Distributional Representation

For the next experiment, we replace the discrete input features with a continuous space representation by looking up the embedding of each word, and concatenate the embeddings of a five word window centered around the current position. Four binary features are also appended to each word embedding to capture capitalization patterns, as described in Collobert et al. (2011). Results of the CRF and SLNN under this setting for the NER task is show in Table 3.

With a continuous space representation, the SLNN model works significantly better than a CRF, by as much as 7% on the CoNLL development set, and 3.7% on ACE dataset. This suggests that there exist statistical dependencies within this low-dimensional (300) data that cannot be effectively captured by linear transformations, but can be modeled in the non-linear neural nets. This perhaps coincides with the large performance im-

	CoNLL <sub>d</sub>	CoNLL <sub>t</sub>	ACE	MUC
CRF <sub>discrete</sub>	90.7	85.0	77.4	73.5
CRF <sub>join</sub>	92.4	87.7	82.2	81.1
SLNN <sub>continuous</sub>	86.6	80.7	75.0	70.8
SLNN <sub>join</sub>	91.9	87.1	81.2	79.7

Table 4: Results of CRF and SLNN when word embeddings are appended to the discrete features. Numbers shown are F1 scores.

provements observed from neural nets in handwritten digit recognition datasets as well (Peng et al., 2009; Do and Artieres, 2010), where dimensionality is also relatively low.

## 5.3 Combine Discrete and Distributional Features

When we join word embeddings with discrete features, we see further performance improvements, especially in the out-of-domain datasets. The results are shown in Table 4.

A similar effect was also observed in Turian et al. (2010). The performance of both the CRF and SLNN increases by similar relative amounts, but the CRF model maintains a lead in overall absolute performance.

## 6 Conclusion

We carefully compared and analyzed the non-linear neural networks used in Collobert et al. (2011) and the widely adopted CRF, and revealed their close relationship. Through extensive experiments on NER and Syntactic Chunking, we have shown that non-linear architectures are effective in low dimensional continuous input spaces, but that they are not better suited for conventional high-dimensional discrete input spaces. Furthermore, both linear and non-linear models benefit greatly from the combination of continuous and discrete features, especially for out-of-domain datasets. This finding confirms earlier results that distributional representations can be used to achieve better generalization.

## Acknowledgments

The authors would like to thank Rob Voigt, Sida Wang, and the three anonymous reviewers, and acknowledge the support of the DARPA Broad Operational Language Translation (BOLT) program through IBM. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA or the US government.

## References

- Rie K. Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853.
- Jean-Gabriel Attali and Gilles Pagés. 1997. Approximations of functions by a multilayer perceptron: a new approach. *Neural Networks*, 10:1069–1081.
- Yoshua Bengio and Samy Bengio. 2000. Modeling high-dimensional discrete data with multi-layer neural networks. In *Proceedings of NIPS 12*.
- Yoshua Bengio. 2009. Learning deep architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, January.
- Francesco Camastra. 2003. Data dimensionality estimation methods: A survey. *Pattern Recognition*, 36:2945–2954.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*.
- Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2461–2505.
- Trinh-Minh-Tri Do and Thierry Artieres. 2010. Neural conditional random fields. In *Proceedings of AIS-TATS*.
- Jenny R. Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL*.
- Thorsten Joachims. 2004. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer Academic Publishers.
- Percy Liang, Hal Daume, and Dan Klein. 2008. Structure compilation: Trading structure for features. In *Proceedings of ICML*.
- Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45:503–528.
- Jian Peng, Liefeng Bo, and Jinbo Xu. 2009. Conditional neural fields. In *Proceedings of NIPS 22*.
- Xaq Pitkow. 2012. Compressive neural representation of sparse, high-dimensional probabilities. In *Proceedings of NIPS 25*.
- Rohit Prabhavalkar and Eric Fosler-Lussier. 2010. Backpropagation training for multilayer conditional random field based phone recognition. In *Proceedings of ICASSP*.
- Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of CoNLL*.
- 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 EMNLP*.
- Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*.
- Michel Verleysen, Damien Francois, Geoffroy Simon, and Vincent Wertz. 2003. On the effects of dimensionality on data analysis with neural networks. In *Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks: Part II*.



# Case Study of Model Adaptation: Transfer Learning and Online Learning

Kenji Imamura

NTT Media Intelligence Laboratories  
1-1 Hikari-no-oka, Yokosuka, 239-0847 Japan  
imamura.kenji@lab.ntt.co.jp

## Abstract

Many NLP tools are released as programs that include statistical models. Unfortunately, the models do not always match the documents that the tool user is interested in, which forces the user to update the models.

In this paper, we investigate model adaptation under the condition that users cannot access the data used in creating the original model. Transfer learning and online learning are investigated as adaptation strategies. We test them on the category classification of Japanese newspaper articles. Experiments show that both transfer and online learning can appropriately adapt the original model if the dataset for adaptation contains all data, not just the data that cannot be well handled by the original model. In contrast, we confirmed that the adaptation fails if the dataset contains only erroneous data as indicated by the original model.

## 1 Introduction

Recent natural language processing (NLP) systems are built using machine learning (supervised learning). The developers of these systems basically create annotated corpora from which statistical models are generated. However, if the documents that users want to apply the systems to do not belong to the domain of the annotated corpora, the resulting accuracy tends to be unsatisfactory.

For instance, Figure 1 shows the typical drop in accuracy in the category classification task of newspaper articles over time; the statistical model was trained using supervised data from 1995 (details are described later). Even though the test data were obtained from newspaper articles (i.e., the same domain data), the accuracy against 2007

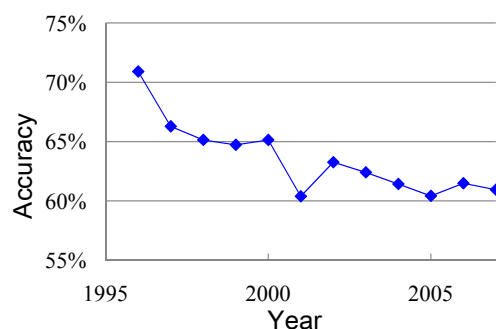


Figure 1: Accuracy of Category Classification with training by 1995 Dataset

articles fell by about 10% from the 1996 articles. The reasons for this include the emergence of new words and changes in word distribution. In order to recover this degradation, we have to re-train the models.

To overcome this problem, transfer learning methods have been proposed (Pan and Yang, 2010). Many transfer learning methods assume that the users can obtain both the original data and additional data for adaptation. However, in most practical cases, the users sometimes are unable to access the original data. For example, only the developers are licensed to handle the original data, not the users.

NLP tools, such as taggers, parsers, and classifiers, are commonly released as programs that include the original models. Since many users cannot update the original models, they continue to use them even if the user's documents do not match the models (Figure 2).

The objective of this paper is to investigate methods that, given an additional dataset, permit adaptation of original models under the constraint that the original dataset is unavailable.

The target task of this paper is category classification of newspaper articles. Because NLP tools such as taggers or parsers are founded on struc-

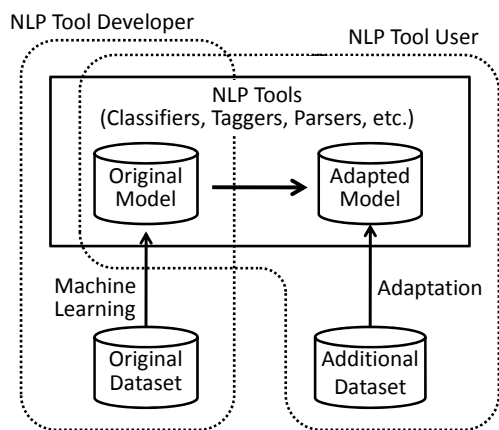


Figure 2: Relationship between Developers and Users of NLP Tools

tured learning, which extends the classification, we select the linear classification task.

In this paper, we investigate the combination of the following learning methods and additional datasets.

- We test two learning methods, batch and online learning. In batch learning, we use a maximum entropy classifier (Berger et al., 1996; Chen and Rosenfeld, 2000) and adapt the model using transfer learning. In online learning, we select soft confidence-weighted learning (Wang et al., 2012).
- We test two kinds of additional datasets. One is that all data are used for adaptation. The other is that only the data that failed to predict correct categories by the original model are used. We consider the active learning strategy for the second dataset.

The remainder of this paper is organized as follows. In Section 2, we detail the task, datasets, and learning methods (batch and online learning). Section 3 describes the experiments conducted and their results, and Section 4 summarizes the findings of this study.

## 2 Settings and Adaptation Methods

### 2.1 Task and Data

The task of this study is category classification of Japanese newspaper articles. We selected articles from Mainichi Shinbun newspapers for the years of 1995, 1996, 2005, and 2006. A part of the 1995 data is widely used in the Japanese

Set Name	Period	# of Data
Original Dataset	Jan.,1995 - Nov.,1995	102,454
Additional Dataset	Jan.,2005 - Nov.,2005	88,202
Development set	Dec.,1995	9,043
Test set 1	Jan.,1996 - Dec.,1996	114,116
Test set 2	Jan.,2006 - Dec.,2006	95,761

Table 1: Statistics of Data Used

NLP community because its dependency structures and predicate-argument structures have been annotated<sup>1</sup>.

One of 16 categories is assigned to each article. The category denotes type of the article, such as ‘Economics’, ‘International’, ‘Sports’, ‘Top page’, and so on. The task of this study is to predict the category of each article from its content (text).

Figure 3 shows the relationships among datasets (for learning and testing) and models. We took articles from Jan. to Nov. in 1995 as the original dataset, and used them to train the original model. The original dataset was not used thereafter. Articles from Dec. 1995 were used to tune the model’s hyperparameters. The additional dataset for adaptation was created from articles from Jan. to Nov. 2005. We prepared two test sets. The first consisted of 1996 articles (Test set 1), and the second consisted of 2006 articles (Test set 2). Our objective is to improve the accuracy against Test set 2. The statistics of the datasets are shown in Table 1.

Features for classification are ‘bag-of-words’ of the title and the first paragraph of the article. Only content words (nouns, verbs, adverbs, adjectives, and interjections) that appear more than once are used as features.

### 2.2 Transfer Learning from Batch Learning

#### 2.2.1 Regularized Adaptation

The problem setting of this paper is a sort of transfer learning (domain adaptation). Because we cannot access the original data, this problem is regarded as “model-based domain adaptation” according to the taxonomy of transfer learning by Sha and Kingsbury (2012). Regularized adaptation (Evgeniou and Pontil, 2004; Xiao and Bilmes, 2006) is a variant of model-based domain adaptation. As the regularizer, it uses the differences

<sup>1</sup>Dependency structures are published as Kyoto University Text Corpus ([http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?Kyoto University Text Corpus](http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?Kyoto%20University%20Text%20Corpus)). Predicate-argument structures are published as NAIST Text Corpus (Iida et al., 2007) (<http://cl.naist.jp/nldata/corpus/>). Note that the texts of the articles must be purchased from the newspaper company.

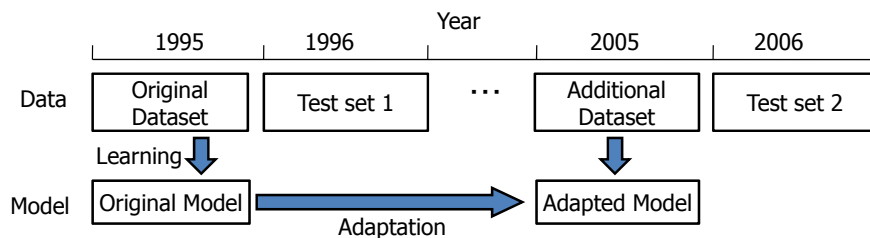


Figure 3: Datasets and Models

in parameters between the adapted model and the original model, not adapted parameters. This is done to minimize the differences between the original model and the adapted model.

Although Evgeniou and Pontil (2004) proposed regularized adaptation for SVMs, and Xiao and Bilmes (2006) proposed the same for neural networks, they can also be applied to maximum entropy classifiers<sup>2</sup>. The loss function,  $\ell$ , is represented as follows.

$$\ell = - \sum_i \log P(y_{AD_i} | \mathbf{x}_{AD_i}; \mathbf{w}_{AD}) + \frac{1}{2C} \sum_{k=1}^d (w_{AD_k} - w_{OR_k})^2, \quad (1)$$

where  $P(y|\mathbf{x}; \mathbf{w})$  denotes the posterior probability of a sample computed with the weight parameters of the model  $\mathbf{w}$ ;  $y_{AD_i}$  and  $\mathbf{x}_{AD_i}$  are the input and output of the  $i$ th sample in the additional dataset, respectively,  $w_{AD_k}$  and  $w_{OR_k}$  denote weight parameters of the adapted and the original model, respectively; both have dimensions of  $d$ , and  $C$  is a hyperparameter.

The maximum entropy classifier used in this paper estimates the weight parameters to minimize the above loss function. The first term in Equation (1) suppresses discriminative errors of the additional data at minimum, and the second term suppresses differences between the original model and the adapted model.

### 2.2.2 Regularization with Two Hyperparameters

The output classes of the adapted model are identical to those of the original model in this task. In contrast, features for classification are not identical because new words appear over time.

<sup>2</sup>Regularized adaptation is used as a re-training function of the Japanese morphological analyzer MeCab (Kudo et al., 2004), which is based on conditional random fields (CRFs). <http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>

In Equation (1), all features, which include features from the original data and the additional data, are treated equally. However, if we significantly change weight parameters of the original model features, the original model can correctly classify less data due to errors. In contrast, with regard to the new features from the additional data, we can change the parameters without limitation. Therefore, it is natural to distinguish new features from those of the original model.

Here, assuming that the number of dimensions of the parameters in the original model is  $d_{OR}$ , and that in the adapted model (i.e., the features include the original and additional data) is  $d_{AD}$ , the loss function becomes,

$$\ell = - \sum_i \log P(y_{AD_i} | \mathbf{x}_{AD_i}; \mathbf{w}_{AD}) + \frac{1}{2C_{AD}} \sum_{k=1}^{d_{OR}} (w_{AD_k} - w_{OR_k})^2 + \frac{1}{2C_{OR}} \sum_{k=d_{OR}+1}^{d_{AD}} w_{AD_k}^2, \quad (2)$$

where  $C_{OR}$  denotes the hyperparameter that was used while learning the original model, and  $C_{AD}$  denotes the hyperparameter for the additional data. If we set them as  $C_{OR} \geq C_{AD}$ , only the new features from the additional data can change significantly; changes to the existing features of the original model are suppressed.

### 2.3 Online Learning

Online learning is a strategy that updates current parameters in order to correctly classify training samples one-by-one. It matches the problem setting in this paper because it can train a new model by altering the original model to suit the additional data. However, it usually loses information about old samples (in our case, the original data). Therefore, we need to iterate the learning process on the entire dataset several times.

The recent proposal Confidence-weighted learning (CW) generates each weight parameter from a Gaussian distribution whose mean is  $\mu$  and standard deviation is  $\sigma$  (Dredze et al., 2008; Crammer et al., 2009a). This method expresses confidence in frequently updated parameters, and accepts only small changes to them. Rarely updated parameters can be greatly changed. Confidence is expressed by a covariance matrix. CW is known to offer faster convergence than the conventional online learning algorithms such as perceptrons and passive-aggressive methods. In other words, CW makes learned samples hard to forget. The CW algorithm offers the possibility of adapting to the additional data without referring to the original data.

It is known that the training performance of the original CW algorithm suffers if the training samples contain significant noise components that are linearly-inseparable. The adaptive regularization of weight algorithm (AROW; (Crammer et al., 2009b)) and the soft confidence-weighted learning algorithm (SCW; (Wang et al., 2012)) were proposed to overcome this weakness. In this paper, we employ the SCW algorithm.

In SCW-I, which uses a linear penalty, parameter updating is represented as follows.

$$\begin{aligned}
 (\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = & \\
 \arg \min_{\boldsymbol{\mu}, \Sigma} \{ & D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \| \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)) + \\
 C \ell^\phi(\mathcal{N}(\boldsymbol{\mu}, \Sigma), & (\boldsymbol{x}_t, y_t)) \}, \quad (3)
 \end{aligned}$$

where  $\boldsymbol{\mu}$  denotes a mean vector and  $\Sigma$  denotes a covariance matrix of the parameters,  $D_{KL}(\cdot \| \cdot)$  denotes Kullback-Leibler divergence,  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$  denotes a multivariate normal distribution with mean of  $\boldsymbol{\mu}_k$  and standard deviation of  $\sigma_k$ ,  $\ell^\phi(\cdot)$  is a loss function based on the hinge loss, and  $C$  is a hyperparameter that restricts the maximum change permitted in the update. Following Equation (3), the loss of the correct class  $y_t$  predicted from input feature vector  $\boldsymbol{x}_t$  becomes minimum by the second term, and simultaneously the change in parameters is suppressed by the first term. (Final update formulae are provided in (Wang et al., 2012)).

However, there are some problems in implementing Equation (3) directly. The following approximations are applied in general.

- Weight parameters  $w$  should be generated from Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ , but the mean vector  $\boldsymbol{\mu}$  is directly used as weight parameters.

- The size of the covariance matrix is  $d \times d$ , where  $d$  denotes the number of dimensions of the parameters, and so memory consumption is high. To avoid this problem, only diagonal elements are considered (the matrix is degenerated to a vector).

In addition, the hyperparameters that control the maximum change and the confidence value,  $C$  and  $\phi$ , must be set manually.

To apply the SCW algorithm, we first construct the original model from the original data using Equation (3) until classification errors become minimum on the development set. Note that the original model retains not only the mean vector but also the covariance matrix. In adaptation, we regard the original model as  $(\boldsymbol{\mu}_0, \Sigma_0)$  and similarly update it using the additional data, one-by-one.

### 3 Experiments

#### 3.1 Experimental Settings

**Methods** We test the methods described in Sections 2.2 and 2.3 (represented as ‘Transfer’ and ‘Online,’ respectively). The following baselines are also tested.

- (a) Original Model. This case yields the upper bound of Test set 1.
- (b) The model is trained using only the additional dataset. If there is enough data, this yields the upper bound of Test set 2.
- (c) The model is trained by the feature augmentation method (Daumé, 2007) using the original data and the additional data, which is one of the domain adaptation techniques. This case yields the upper bound if we can access the original data.
- (d) The case in which the models of (a) and (b) are interpolated at the ratio of 1:1. This provides a baseline for the lack of access to the original data.

**Additional Datasets** We used two types of additional datasets. One is (e) all data in 2005 newspapers are used for adaptation (Normal Case). The other is (f) only the data unknown to the original model are used (Active Learning). In practical cases, we want to adapt the model when we find a failure of the original model. Therefore, case (f) is a practical setting. The additional datasets of the original models have different numbers of

Type	Method/Dataset	Transfer		Online	
		Test1	Test2	Test1	Test2
Baselines	(a) Original Model	<b>70.90%</b>	61.49%	<b>71.41%</b>	62.60%
	(b) Additional Only	56.73%	<b>75.66%</b>	57.07%	<b>76.36%</b>
	(c) Original + Additional Data	<b>70.99%</b>	<b>75.77%</b>	<b>72.00%</b>	<b>76.58%</b>
	(d) Interpolation	68.70%	72.28%	68.49%	72.98%
Model Adaptation	(e) Normal Case	64.26%	<b>75.78%</b>	66.87%	<b>75.78%</b>
	(f) Active Learning	50.32%	63.29%	57.28%	65.81%

Table 2: Test Set Accuracies of Methods and Datasets

entities, 34,950 and 33,633 for the Transfer and Online cases, respectively.

**Tuning** The hyperparameters are optimized against the development set when the original models are trained and the same values are used in all experiments.

### 3.2 Results of the Methods

The results are shown in Table 2.

First of all, focusing on baselines (a) and (b), Test set 1 yielded basically the highest accuracies for case (a), while for (b) it was Test set 2. Using datasets that are near to the test sets yields better model training in this task.

Focusing on case (c), in which training uses both the original and the additional datasets, the advantages of cases (a) and (b) are secured. However, although we applied domain adaptation, the improvements from cases (a) and (b) were little. This result indicates that the size of the additional dataset was sufficient and that the model matched the upper bound by using just the additional dataset. In addition, we confirmed that the accuracies of interpolation (d) were intermediate between those of (a) and (b).

While accuracy slightly differed with the learning method, the Transfer and Online cases exhibited the same tendency.

Next, for normal case (e) in model adaptation, both Transfer and Online achieved basically the highest accuracies against Test set 2. This result shows that model adaptation worked effectively. On the other hand, focusing on the accuracies of Test set 1, Online learning exhibited a smaller degradation from the original model (a) than Transfer. We suppose that this difference is due to the difference between maximum entropy and SCW-I, rather than that between the transfer/online learning. The maximum entropy method optimizes parameters based on the maximum a posteriori (MAP), and it is sensitive to probability distribution. In contrast, SCW-I used

in Online is based on margin criteria, and ignores data outside the margin. Therefore, Online yielded smaller degradation.

In the case of active learning (f), the effects of model adaptation were little compared to the other cases. Namely, improvements against Test set 2 were slight and the accuracies of Test set 1 were degraded from the original model (a). Because transfer learning assumes that the target domain should be similar to the source domain, the dataset difference impacts performance significantly. We can conclude that we should collect (and use) all data for model adaptation regardless of whether or not the original model can correctly classify it.

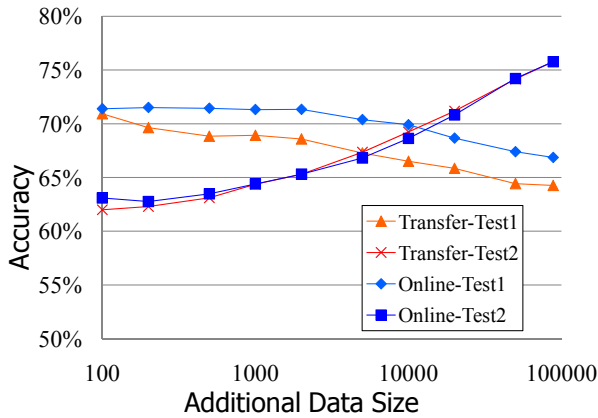
### 3.3 Accuracies according to Additional Data Size

Figure 4 plots accuracy versus the size of the additional datasets. In the normal case (e), the accuracies of Test set 2 improved with both the Transfer and the Online cases along with dataset size. In contrast, the accuracy of Test set 1 with Transfer degraded faster than Online, as described in Section 3.2. The degradation with Online started when over 2,000 data points were added. This result shows that the SCW-I algorithm of Online is relatively robust and remembers the previously learnt data.

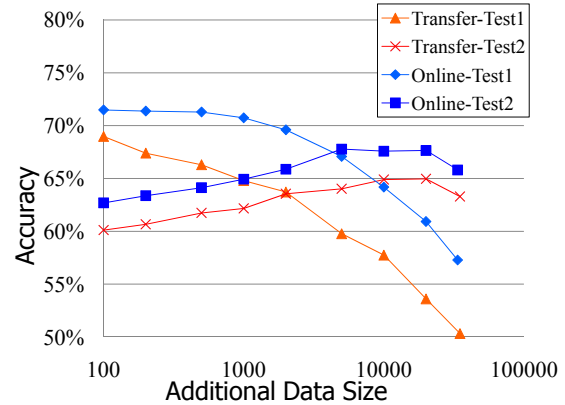
Focusing on active learning (f), the accuracies of Test set 2 were degraded with both Transfer and Online when all additional data was used. The addition of huge amounts of erroneous data causes a harmful effect regardless of the learning method used.

### 3.4 Hyperparameters in Transfer Learning

Finally, Table 3 shows the accuracies when the hyperparameters for the existing features in the original model and the new features that appear only in the additional data were distinguished by the method described in Section 2.2.2. Here, hyperparameter  $C_{OR}$  was set when the original model



(e) Model Adaptation (Normal Case)



(f) Model Adaptation (Active Learning)

Figure 4: Test Set Accuracy versus Size of Additional Data

Method	$C_{OR}$	$C_{AD}$	Test 1	Test 2
(e) Normal Case	0.1	0.001	70.33%	67.26%
	0.1	0.01	68.09%	72.71%
	0.1	0.1	64.26%	75.78%
(f) Active Learning	0.1	0.001	67.70%	65.15%
	0.1	0.01	61.28%	66.87%
	0.1	0.1	50.32%	63.29%

Table 3: Accuracies of Different Hyperparameters

was trained, and only  $C_{AD}$  was changed.

In the normal case, while the changes to the existing parameters were suppressed (small  $C_{AD}$ ), the accuracy of Test set 2 decreased. However, it was higher than that of the original model (61.49%  $\rightarrow$  67.26%), and the accuracy on Test set 1 was almost constant (70.90%  $\rightarrow$  70.33%). If we have to adapt the model under the condition that the original performance is to be maintained, the two hyperparameter approach is effective.

In the active learning case, although we distinguished the hyperparameters, the results were not improved from the normal case.

## 4 Conclusions

We investigated the characteristics of model adaptation wherein the original training data cannot be accessed. We tested transfer learning (regularized adaptation) on the maximum entropy classifier and online learning (soft confidence-weighted learning). Our results are summarized as follows.

- If the additional dataset contains all data, regardless of whether it can be correctly classified by the original model or not, both transfer learning and online learning basically achieved the highest accuracy.

- However, the maximum entropy classifier with regularized adaptation changed more data, which the original model correctly classified, yielding more errors than online learning by SCW-I.

- Restricting the additional data to the data that the original model could not classify correctly had negative effects in our problem setting (i.e., the original dataset cannot be accessed).

- We could slightly adapt the model while retaining previous classification performance by distinguishing the hyperparameters for the existing features and those for the new features.

In natural language processing, structured learning is frequently used for sequential labeling, parsing, and so on. Our future work is to apply model adaptation to structured learning.

## References

- Adam L. Berger, Stephan A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.
- Stanley F. Chen and Ronald Rosenfeld. 2000. A survey of smoothing techniques for maximum entropy models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.
- Koby Crammer, Mark Dredze, and Alex Kulesza. 2009a. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 496–504, Singapore.
- Koby Crammer, Alex Kulesza, and Mark Dredze. 2009b. Adaptive regularization of weight vectors.

- In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 414–422.
- Hal Daumé, III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 256–263, Prague, Czech Republic.
- 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, New York, NY, USA. ACM.
- Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pages 109–117.
- Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 132–139, Prague, Czech Republic.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain.
- Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, 22(10):1345–1359, October.
- Fei Sha and Brian Kingsbury. 2012. Domain adaptation in machine learning and speech processing. Interspeech 2012 Tutorial. <http://www-bcf.usc.edu/~feisha/pubs/IS2012Tutorial.pdf>.
- Jialei Wang, Peilin Zhao, and Steven C. Hoi. 2012. Exact soft confidence-weighted learning. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 121–128, New York, NY, USA. ACM.
- Li Xiao and Jeff Bilmes. 2006. Regularized adaptation of discriminative classifiers. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing Volume I*, pages 237–240.

# Source and Translation Classification using Most Frequent Words

**Zahurul Islam**

AG Texttechnology

Institut für Informatik

Goethe-Universität Frankfurt

zahurul@em.uni-frankfurt.de

**Armin Hoenen**

AG Texttechnology

Institut für Informatik

Goethe-Universität Frankfurt

hoenen@em.uni-frankfurt.de

## Abstract

Recently, translation scholars have made some general claims about translation properties. Some of these are source language independent while others are not. Koppel and Ordan (2011) performed empirical studies to validate both types of properties using English source texts and other texts translated into English. Obviously, corpora of this sort, which focus on a single language, are not adequate for claiming universality of translation properties. In this paper, we are validating both types of translation properties using original and translated texts from six European languages.

## 1 Introduction

Even though it is content words that are semantically rich, function words also play an important role in a text. Function words are more frequent and predictable than content words. Generally, function words carry grammatical information about content words. High frequency function words are relatively shorter than mid/low frequency function words (Bell et al., 2008). Due to their high frequency in texts and their grammatical role, function words also indicate authorial style (Argamon and Levitan, 2005). These words could play an important role in translated text and in the translation process.

Source and translation classification is useful for some Natural Language Processing (NLP) applications. Lembersky et al. (2011) have shown that a language model from translated text improves the performance of a Machine Translation (MT) system. A source and translation classifier

can be used to identify translated text. This application also can be used to detect plagiarism where the plagiarised text is translated from another language.

From the early stage of translation studies research, translation scholars proposed different kinds of properties of source text and translated text. Recently, scholars in this area identified several properties of the translation process with the aid of corpora (Baker, 1993; Baker, 1996; Olohan, 2001; Laviosa, 2002; Hansen, 2003; Pym, 2005). These properties are subsumed under four keywords: *explicitation*, *simplification*, *normalization* and *levelling out*. They focus on the general effects of the translation process.

Toury (1995) has a different theory from these. He stated that some *interference* effects will be observable in the translated text. That is, a translated text will carry some fingerprints of its source language. Specific properties of the English language are visible in user manuals that have been translated to other languages from English (for instance, word order) (Lzwaini, 2003). Recently, Pastor et al. (2008) and Ilisei et al. (2009; 2010) have provided empirical evidence of simplification translation properties using a comparable corpus of Spanish.

Koppel and Ordan (2011) perform empirical studies to validate both theories, using a sub-corpus extracted from the *Europarl* (Koehn, 2005) and IHT corpora (Koppel and Ordan, 2011). They used a comparable corpus of original English and English translated from five other European languages. In addition, original English and English translated from Greek and Korean was also used in their experiment. They have found that a translated text contains both source language dependent and independent features.



Obviously, corpora of this sort, which focus on a single language (e.g., English), are not adequate for claiming the universal validity of translation properties. Different languages (and language families) have different linguistic properties. A corpus that contains original and translated texts from different source languages will be ideal for this kind of study. In this paper, we are validating both types of translation properties using original and translated texts from six European languages. As features, we used frequencies of the 100 most frequent words of each target language.

The paper is organized as follows: Section 2 discusses related work, followed by an introduction of our corpus in Section 3. The experiment and evaluation in Section 4 are followed by a discussion in Section 5. Finally, we present conclusions and future work in Section 6.

## 2 Related Work

Corpus-based translation studies is a recent field of research with a growing interest within the field of computational linguistics. Baroni and Bernardini (2006) started corpus-based translation studies empirically, where they work on a corpus of geo-political journal articles. A Support Vector Machine (SVM) was used to distinguish original and translated Italian text using n-gram based features. According to their results, word bigrams play an important role in the classification task.

Van Halteren (2008) uses the *Europarl* corpus for the first time to identify the source language of text for which the source language marker was missing. Support vector regression was the best performing method.

Pastor et al. (2008) and Ilisei et al. (2009; 2010) perform classification of Spanish original and translated text. The focus of their works is to investigate the *simplification* relation that was proposed by (Baker, 1996). In total, 21 quantitative features (e.g. a number of different POS, average sentence length, the parse-tree depth etc.) were used where, nine (9) of them are able to grasp the simplification translation property.

Koppel and Ordan (2011) have built a classifier that can identify the correct source of the translated text (given different possible source languages). They have built another classifier which can identify source text and translated text. Furthermore, they have shown that the degree of difference between two translated texts, translated

from two different languages into the same target language reflects, the degree of difference of the source languages. They have gained impressive results for both of the tasks. However, the limitation of this study is that they only used a corpus of English original text and English text translated from various European languages. A list of 300 function words (Pennebaker et al., 2001) was used as feature vector for these classifications.

Popescu (2011) uses *string kernels* (Lodhi et al., 2002) to study translation properties. A classifier was built to classify English original texts and English translated texts from French and German books that were written in the nineteenth century. The *p-spectrum* normalized kernel was used for the experiment. The system works on a character level rather than on a word level. The system performs poorly when the source language of the training corpus is different from the one of the test corpus.

We can not compare our findings directly with Koppel and Ordan (2011) even though we use text from the same corpus and similar techniques. The English language is not considered for this study due to unavailability of English translations for some languages included in this work. Furthermore, instead of the list of 300 function words used by Koppel and Ordan (2011), we used the 100 most frequent words for each candidate language.

## 3 Data

The field of translation studies lacks a multilingual corpus that can be used to validate translation properties proposed by translation scholars. There are many multilingual corpora available used for different NLP applications. A customized version of the *Europarl* corpus (Islam and Mehler, 2012) is freely available for corpus-based translation studies. However, this corpus is not suitable for the experiment we are performing here. We extract a suitable corpus from the *Europarl* corpus in a way similar to Lembersky et al. (2011) and Koppel and Ordan (2011). Our target is to extract texts that are translated from and to the languages considered here. We trust the source language marker that has been put by the respective translator, as did Lembersky et al. (2011) and Koppel and Ordan (2011).

To experiment with stylistic differences in translated text, a list of function words and their

	German	Dutch	French	Spanish	Polish	Czech
German	-	2,574,110	4,757,076	2,035,736	584,114	215,212
Dutch	4,881,949	-	4,386,270	2,682,935	446,702	149,235
French	5,241,411	659,001	-	2,724,897	659,001	226,435
Spanish	4,020,898	1,925,157	3,696,393	-	662,718	247,219
Polish	451,357	112,274	695,360	194,724	-	82,312
Czech	378,300	105,058	684,061	187,236	214,959	-

Table 1: The customized corpus for source language identification (number of words per language)

respective native frequencies is necessary. Since for many languages such a list does not exist, we pursue an alternative strategy. A list of the 100 most frequent words is available for many languages and since at the same time the majority of these first 100 most frequent words of any language are function words, we use these lists. The 100 most frequent German words are taken from the *Deutscher Wortschatz*.<sup>1</sup> The most frequent Czech word list is taken from the freely available Czech national corpus.<sup>2</sup> The 100 most frequent Spanish words are taken from the book *A Frequency Dictionary of Spanish: Core Vocabulary for Learners* (Davies, 2006). The French most frequent words are taken from the *A Frequency Dictionary of French: Core Vocabulary for Learners* (Lonsdale and Bras, 2009). The 100 most frequent Dutch words are taken from snowball.<sup>3</sup> The most frequent Polish word list are collected from the Polish scientific publisher PWN.<sup>4</sup>

## 4 Experiment

In order to validate two different kinds of translation properties mentioned in Section 1, two different experiments will be performed. For the first experiment, our hypothesis is that texts translated into the same language from different source languages have different properties, a trained classifier will be able to classify texts based on different sources. Our second hypothesis is that translated texts are distinguishable from source texts; a classifier can be trained to identify translated and original texts. Note that we use the Naive Bayes multinomial classifier (Mccallum and Nigam, 1998) in WEKA (Hall et al., 2009) for classification. To overcome the data over-fitting problem, we randomly generate training and test set  $N$  times and calculate the weighted average of  $F$ -Score and  $Ac$ -

<sup>1</sup><http://wortschatz.informatik.uni-leipzig.de/>

<sup>2</sup><http://ucnk.ff.cuni.cz/syn2005.php>

<sup>3</sup><http://snowball.tartarus.org/algorithms/dutch/stop.txt>

<sup>4</sup>[http://korpus.pwn.pl/stslow\\_en](http://korpus.pwn.pl/stslow_en)

	German	Dutch	French	Spanish	Polish	Czech
German	-	197	197	198	201	197
Dutch	197	-	197	198	198	191
French	148	147	-	148	149	157
Spanish	148	147	148	-	148	148
Polish	151	141	149	148	-	129
Czech	140	164	149	148	151	-

Table 2: Source language identification corpus (chunks)

*curacy*. In this experiment the value of  $N$  is 100. The randomly generated training sets contain 80% of the data while the remaining data is used as a test set. To evaluate the classification results, we use standard  $F$ -Score and  $Accuracy$  measures.

### 4.1 Source Language Identification

In this experiment, our goal is to validate the translation properties postulated by Toury (1995). He stated that a translated text inherits some fingerprints from the source language. The experimental result of Koppel and Ordan (2011) shows that text translated into English holds this property. If this characteristic also holds for text translated into other languages, then it will corroborate the claim by Toury (1995). If it does not hold for a single language then it might be claimed that this translation property is not universal. In order to train a classifier, we use texts translated into the same language from different source languages. Table 1 shows the statistics of the corpus used for source language identification experiments. Later, each corpus is divided into a number of chunks (see Table 2). Each chunk contains at least seven sentences. Our hypothesis is again similar to Koppel and Ordan (2011), that is, if the classifier’s accuracy is close to 20%, then we cannot say that there is an *interference* effect in translated text. If the classifier’s accuracy is close to 100% then our conclusion will be that *interference* effects exist in translated text. Table 3 and Table 4 show the evaluation results. Table 3 shows the  $F$ -Scores for translated text from different source languages. Rows represent translated texts and columns represent source languages.

A first minor observation can be made, in that the consistency of the results increases when analyzing them with respect to the concept of *language family*. The term *language family* is broadly used in linguistics as a denomination of groups of languages that have descended from a common

	German	Dutch	French	Spanish	Polish	Czech
German	-	0.97	0.95	0.95	0.80	0.72
Dutch	0.90	-	0.90	0.89	0.62	0.67
French	0.96	0.96	-	0.95	0.78	0.71
Spanish	0.95	0.96	0.87	-	0.74	0.69
Polish	0.53	0.41	0.61	0.48	-	0.49
Czech	0.47	0.36	0.54	0.39	0.67	-

Table 3: Source language identification evaluation (F-Score)

Translated Text	Accuracy
German	88.2%
Dutch	81.1%
French	87.4%
Spanish	84.7%
Polish	51.3%
Czech	50.5%

Table 4: Source language identification evaluation (Accuracy)

ancestor. In the vast majority of cases, members of the same language family share a considerable number of words and grammatical structures. In the experiment, we consider three language families: Romance languages (French and Spanish), Germanic languages (German and Dutch), and Slavic languages (Polish and Czech).

With a Romance target language,<sup>5</sup> the identification of other Romance and of Germanic languages as translation sources performs high, with an F-Score of between 0.86 and 0.95. However, a noticeable drop in performance concerns the identification of the Slavic languages.

When we take a look at the confusion matrices for the respective classifications, we find that, for instance, most misclassifications in the French target language data are between the sources of Polish and Czech. For Germanic target languages, the pattern repeats: when translated into German or Dutch, Polish and Czech texts are hardest to identify as the correct source.

The Slavic target languages show a different pattern. Even in another Slavic target language, a Slavic source language cannot reliably be identified in our setting. In addition to this, translations into Slavic are harder to distinguish from each other. Misclassifications in this case show language family specific patterns: German is, for instance, most often misclassified as Dutch in both the Czech and the Polish data.

<sup>5</sup>Target language refers to text translated into the language

## 4.2 Source Translation Classification

Translated texts have distinctive features that make them different from original or non translated text. According to Baker (1993; 1996), Olohan (2001), Lavisoa (2002), Hansen (2003), and Pym (2005) there are some general properties of translations that are responsible for the difference between these two text types. Some of these properties are source and target language independent. According to their findings, a translated text will be similar to another translated text but will be different from a source text. In the past, researchers have used comparable corpora to validate these translation properties (Baroni and Bernardini, 2006; Pastor et al., 2008; Ilisei et al., 2009; Ilisei et al., 2010; Koppel and Ordan, 2011). Most of them used comparable corpora for two-class classification, distinguishing translated texts from the original texts. Only Koppel and Ordan (Koppel and Ordan, 2011) used English texts translated from multiple source languages. We perform similar experiments only for six European languages as shown in Table 1. In this experiment, the translated text in our training and test set will be a combination of all languages other than the target language. For example: when the original class contains original texts (source) in German, then the translation class contains texts that are translated German texts, translated from French, Dutch, Spanish, Polish, and Czech texts. Each class contains 200 chunks of texts, where as the translated class has 40 chunks from each of the source languages. The source language texts are extracted for the corresponding languages in a similar way from the Europarl corpus. Koppel and Ordan (2011) received the highest accuracy (96.7%) among all works noted above. The training and test data are generated in similar ways as in our previous experiment. That is, 80% of the data is randomly extracted for training and the rest of the data is used for testing. Expected F-Scores are calculated from 100 samples. Table 5 shows the evaluation results. Even though the classifier for German achieves around 99% accuracy, we cannot compare the result with Koppel and Ordan (Koppel and Ordan, 2011) as the amount of chunks for the classes are different. The classifiers for other languages also display very high accuracy.

The result of Table 5 shows that general translation properties exist for all languages used in this experiment.

Language	Accuracy	F-Score
German	99.9%	0.99
Dutch	95.1%	0.95
French	81.9%	0.81
Spanish	94.4%	0.94
Polish	93.3%	0.93
Czech	81.1%	0.81

Table 5: Source translation classification

## 5 Discussion

The results show that training a classifier based on the 100 most frequent words of a language is sufficient to obtain interpretable results. We find our results to be compatible with Koppel and Ordan (2011) who used 300 function words. A list of the 100 most frequent words is easily obtainable for a vast number of languages, while lists consisting strictly of function words are rare and cannot be produced without considerable additional effort.

While the 100 most frequent words of a language are sufficient to train a classifier for Germanic or Romance languages, it fails to perform equally well for Slavic languages. Koppel and Ordan (2011) claim that Toury’s (1995) findings of *interference* of a translation hold true; we find the assumption to be too simplistic, since for Slavic text either as a source or target language this statement cannot be supported.

Although function words do exist in all the languages we examined, the language families differ in the degree to which it is necessary to use them. For instance, French lacks a case system (Dryer and Haspelmath, 2011), and makes instead use of prepositions. On the other hand, Polish and Czech most extensively use (inflectional) affixes (Kulikov et al., 2006). Regarding the distribution of word frequencies, for both Polish and Czech, the use of affixes causes a flatter Zipf curve. Kwapien et al. (2010) put it so :“...typical Polish texts have smaller  $\alpha$  [as exponent of the formula  $f(r) \sim r^{-\alpha}$ ] than typical English texts (on average: 0.95 vs. 1.05).” This means that on average a more frequent word does not differ as much in its frequency from a word 10 ranks further down in Polish as it does in English. Consequently, there will be fewer instances of the 100 most frequent words in the same portion of text. This is an obvious reason why a classifier’s training must remain weaker in comparison to languages with a steeper Zipf curve. There is a positive correlation to language family when considering the probabil-

ity of finding the same strategy (e.g. prepositions vs. affixes). In summary, the fact that Slavic uses more affixes, or is more inflectional in linguistic terms, explains to some extent why the classifier performs worst for Slavic target text.

However, for Slavic source texts, the classification results are equally unsatisfactory, which has to be explained differently. One phenomenon contributing here could be that Romance and Germanic have a recent history of mutual loans and calques, which increases the probability of finding synonyms where one has a Romance origin and one a Germanic origin. In the case of a translation, the translator, when confronted with such a synonym, might choose the item similar to the source language within the target language, as this minimizes the translation effort, complies thus to an economy principle and has virtually no effect on the translation.<sup>6</sup> Making this choice, the translator unintentionally distorts the native frequency patterns for the target language. This could be one of the processes generating an imprint of translated text in the frequency spectrum, since function words are also subject to loaning and synonymy.

If the translator has a choice for translating a preposition/affix and neither of the possibilities is similar to the source language, nor a loanword or structurally similar, he/she will go for the predominant word or structure of the target language (since he/she is a native of the target language by translation industry standard), making the translation less different from native text. The data can be influenced by many additional variables such as differing translation paradigms influencing the choice of structures (free translation vs. faithful translation), different industry standards, the size of the chunks,<sup>7</sup> the quality of the translation source marking, the native tongue of the translator(s), the time pressure for delivery, the payment, the membership of all sample languages to the European subbranch of Indo-European languages, the qualities of the lists of the most frequent 100 words, the genre of the Europarl corpus, and possibly many more.

This said, we believe the best hypothesis for the

<sup>6</sup>For French to English translations an example would be the translator’s choice of “intelligent” as a translation for French “intelligent” in a place where “smart” would have been slightly more natural.

<sup>7</sup>Since a short text contains fewer anaphora and thus personal pronouns.

interpretation of the data is that a good classification result is reached firstly for languages with a more isolating structure, since they make less use of affixes and therefore more of function words, and should display steeper Zipf curves. Secondly, the classification result should be better, the more instances the text contains, where the translator for one token (or for one structure) of the source language has the choice between at least two words or structures in the target language with one of those being similar to the source language, the other being different. The number of such instances most probably correlates positively with the degree and quality of language relationship and language contact since the number of cognates, loans and calques does. However, this number can also be “accidentally high” for two unrelated languages when they overlap in grammatical structure. As has been postulated for instance by Croft (2003), languages undergo a cyclic development from structurally more isolating towards agglutinative to inflectional and then back to isolating. When a language is in a state of transition, which practically all languages are, they offer two structural encoding possibilities for one specific grammatical property, e.g., a genitive (for instance, inflectional (an affix) as in *Peter’s house* and isolating (a preposition) as in *the house of Peter*). All languages should share structural properties, since there are only three types and each language has practically at least two.

Corroborating this rather complex hypothesis, we examine data on Bulgarian and Romanian. We take Bulgarian as the target language. The data showed that the classifier classifies Czech text no worse than Dutch or German and only slightly worse than French. When we replace Czech with Bulgarian and Spanish with Romanian in the German target language, the language family dependent pattern gets blurred and the identification of Polish performs quite well, that of Romanian relatively poor, while French is identified reliably. This together with the observation that Romanian is misclassified either as Polish or Bulgarian and Bulgarian is mostly misclassified as Romanian seems to be a strong hint towards the impact of the language specific usage of function words, linguistic structure, and the importance of language contact. Bulgarian and Romanian constitute the core of the most prominent linguistic contact zone or *sprachbund* ever written on the Balkans. This

suggests that Romanian and Bulgarian translators may, due to grammatical convergence of their languages make, given two equivalent structures in any target language, the same (structurally motivated) choices and hence leave a very similar imprint. That is, sprachbund membership as well as language family could be decisive factors for a classifiers performance.

## 6 Conclusion

We have shown that *interference* as originally proposed by Toury (1995) is not supported by the data without making further assumptions. Language family and language contact should be considered separately for each language pair as sources for possible weak results of a classifier even when operating with function words as should be general structural similarity. As for the properties of translated text being universal, we found support for this in our data in a real n-ary validation setting. We have also shown that the much more easily obtainable lists of the 100 most frequent words work almost as well for classification as do longer lists that contain only function words.

## 7 Acknowledgments

We would like to thank Prof. Moshe Koppel and Dr. Noam Ordan for providing their data, and Prof. Dr. Alexander Mehler for suggestions and comments. We also thank Dr. Timothy Price for checking English and three anonymous reviewers. This work is funded by the LOEWE Digital-Humanities project at the Goethe-Universität Frankfurt.

## References

- Shlomo Argamon and Shlomo Levitan. 2005. Measuring the usefulness of function words for authorship attribution. In *The Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*.
- Mona Baker. 1993. Corpus linguistics and translation studies - implications and applications. In Mona Baker, Gill Francis, and Elena Tognini-Bonelli, editors, *Text and Technology. In Honour of John Sinclair*, pages 233–354. John Benjamins.
- Mona Baker. 1996. Corpus-based translation studies: The challenges that lie ahead. In *LSP and Translation: Studies in Language Engineering in Honour of Juan C. Sager*, pages 175–186. Amsterdam & Philadelphia: John Benjamins.

- Marco Baroni and Silvia Bernardini. 2006. A new approach to the study of translationese: Machinelearning the difference between original and translated text. *Literary and Linguistic Computing*, 21(3):259–274.
- Alan Bell, Jason M. Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2008. Predictability effects on durations of content and function words in conversational english. *Elsevier Journal of Memory and Language*, 60:92–111.
- William Croft. 2003. *Typology and Universals*. Cambridge textbooks in linguistics. Cambridge University Press.
- Mark Davies. 2006. *A Frequency Dictionary of Spanish: Core Vocabulary for Learners*. Taylor & Francis.
- Matthew S. Dryer and Martin Haspelmath. 2011. The world atlas of language structures online.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations*, 11(1):10–18.
- Silvia Hansen. 2003. *The Nature of Translated Text: An Interdisciplinary Methodology for the Investigation of the Specific Properties of Translations*. Ph.D. thesis, University of Saarland.
- Iustina Ilisei, Diana Inkpen, Gloria Corpas Pastor, and Ruslan Mitkov. 2009. Towards simplification: A supervised learning approach. In *Proceedings of Machine Translation 25 Years On, London, United Kingdom, November 21-22*.
- Iustina Ilisei, Diana Inkpen, Gloria Corpas Pastor, and Ruslan Mitkov. 2010. *Identification of translationese: A machine learning approach*, pages 503–511. Springer.
- Zahurul Islam and Alexander Mehler. 2012. Customization of the europarl corpus for translation studies. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.
- Moshe Koppel and Noam Ordan. 2011. Translationese and its dialects. In *49th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Leonid Kulikov, Andrej Malchukov, and Peter de Swart, editors. 2006. *Case, Valency and Transitivity*, volume 77 of *Studies in Language Companion Series*.
- Jaroslaw Kwapien, Stanislaw Drozd, and Adam Orczyk. 2010. Linguistic complexity: English vs. polish, text vs. corpus. *CoRR*, abs/1007.0936.
- Sara Laviosa. 2002. *Corpus-based translation studies. Theory, findings, applications*. Amsterdam/New York: Rodopi.
- Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2011. Language models for machine translation: Original vs. translated texts. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.
- Deryle Lonsdale and Yvon Le Bras. 2009. *A Frequency Dictionary of French: Core Vocabulary for Learners*. Routledge.
- Sattar Lzwaini. 2003. Building specialised corpora for translation studies. In *Workshop on Multilingual Corpora: Linguistic Requirements and Technical Perspectives, Corpus Linguistics*.
- Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on 'Learning for Text Categorization'*.
- Maeve Olohan. 2001. Spelling out the optionals in translation:a corpus study. In *Corpus Linguistics 2001 conference. UCREL Technical Paper number 13. Special issue*.
- Gloria Corpas Pastor, Ruslan Mitkov, Naveed Afzal, and Viktor Pekar. 2008. Translation universals: do they exist? a corpus-based NLP study of convergence and simplification. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas (AMTA-08)*.
- Jams W. Pennebaker, Martha E. Francis, and Roger J. Booth. 2001. *Linguistic Inquiry and Word Count (LIWC): LIWC2001 Manual*. Erlbaum Publishers.
- Marius Popescu. 2011. Studying translationese at the character level. In *Recent Advances in Natural Language Processing*.
- Anthony Pym. 2005. Explaining explicitation. In *New Trends in Translation Studies. In Honour of Kinga Klaudy*, pages 29–34. Akademia Kiad.
- Gideon Toury. 1995. *Descriptive Translation Studies and Beyond*. John Benjamins, Amsterdam/Philadelphia.
- Hans van Halteren. 2008. Source language markers in europarl translations. In *International Conference in Computational Linguistics (COLING)*, pages 937–944.

# Comparison of Algorithmic and Human Assessments of Sentence Similarity

**John G. Mersch**

Department of Mathematics  
Xavier University of Louisiana  
New Orleans, LA 70125 USA  
jmersch@xula.edu

**R. Raymond Lang**

Department of Computer Science  
Xavier University of Louisiana  
New Orleans, LA 70125 USA  
rlang@xula.edu

## Abstract

This paper describes a new method, based on information theory, for measuring sentence similarity. The method first computes the information content (IC) of dependency triples using corpus statistics generated by processing the Open American National Corpus (OANC) with the Stanford Parser. We define the similarity of two sentences as a function of (1) the similarity of their constituent dependency triples, and (2) the position of the triples in their respective dependency trees. We apply the algorithm to 15 pairs of sentences that were also given to human subjects to assign a similarity score. The human- and computer-generated scores are compared; the results are promising, but point to the need for further refinement.

## 1 Introduction

This project seeks to develop an algorithm that measures the extent to which the meanings of two given sentences overlap. Our plan is to use such an algorithm in a clustering application (Lang and Mersch, 2012).

The technique described in this paper extends previous work applying an information-theoretic definition of similarity to a number of different domains (Lin, 1998). Lin's information-theoretic definition of similarity performs as well as or better than other information-theoretic similarity metrics that leverage domain specifics (Resnik, 1995; Wu and Palmer, 1994).

The metric being proposed in this paper shares characteristics of word co-occurrence methods and descriptive feature-based methods (Li et al., 2006), in addition to using structural information provided by the Stanford Parser (Klein and Manning, 2003). We test this metric on 15 pairs of sen-

tences, each of which was assessed for similarity by 40 fluent English speakers.

## 2 Background & Related Work

Methods that detect similarity of long documents often utilize co-occurring words (Salton, 1988), since similar texts share a high number of words. But this does not transfer well to short, sentence-length texts, since language allows similar meanings to be expressed using different vocabularies.

Existing text similarity measures suffer from drawbacks. Vector-based methods employ high-dimensional, sparse representations that are computationally inefficient (Landauer et al., 1998; Salton, 1988; Burgess et al., 1998). Some methods rely on extensive manual preprocessing (McClelland and Kawamoto, 1986), making them impractical for large-scale use. Still other methods suffer from domain dependency (Li et al., 2006).

Related work on text similarity may be grouped into three categories:

1. Methods based on word co-occurrence (i.e. "bag of words" methods) disregard the impact of word order on meaning (Meadow et al., 1999); thus, the two sentences:

$T_1$ : The cat killed the mouse.

$T_2$ : The mouse killed the cat.

are regarded as identical, since they use the same words. Documents are represented as vectors in an  $n$ -dimensional space, where  $n$  is the length of a pre-compiled word list, typically in the tens or hundreds of thousands. The resulting representations are sparse and computationally inefficient (Li et al., 2006). Also, these methods often exclude function words (e.g. the, of, an, etc.) that have low relevance for similarity of long documents but convey information important for sentence similarity. These methods will not detect similarity of sentences that use different

words to convey the same meaning. However, they achieve improved results by examining word pairs instead of single words (Okazaki et al., 2003).

2. Corpus-based methods. Latent semantic analysis (LSA) constructs an occurrence count matrix where the rows represent words and the columns text units, usually paragraphs or documents. It is more suitable for longer texts than for sentences (Landauer et al., 1998). Hyperspace Analogues to Language (HAL) (Burgess et al., 1998) constructs a word co-occurrence matrix based on a moving window of a predefined width, typically 10. HAL is also more effective for longer texts than for sentences (Li et al., 2006).
3. Descriptive feature-vector methods. These methods employ pre-defined thematic features to represent a sentence as a vector of feature values, then obtain a similarity measurement through a trained classifier (Taraban and McClelland, 1988). Choosing a suitable set of features and automatically obtaining values for features pose obstacles for these methods (Islam and Inkpen, 2008).

In contrast to the above approaches, Lin (1998) proposes an information-theoretic measure of similarity. This measure is derived from assumptions about similarity rather than from a domain-specific formula. The metric can be applied to any domain with a probabilistic model. From a set of assumptions grounded in information theory, Lin proves a Similarity Theorem:

the similarity between  $A$  and  $B$  is measured by the ratio between the amount of information needed to state the commonality of  $A$  and  $B$  and the information needed to fully describe what  $A$  and  $B$  are:

$$\text{sim}(A, B) = \frac{\log P(\text{common}(A, B))}{\log P(\text{description}(A, B))}$$

[...] If we know the commonality of the two objects, their similarity tells us how much more information is needed to determine what these two objects are. (Lin, 1998)

Lin applies the definition to four different domains; one of these is similarity between words according to the distribution of dependency triples extracted from a text corpus. Lin's test uses a database of 14 million dependency triples extracted from a corpus consisting of items from the *Wall Street Journal* and from the *San Jose Mercury*. He also applies it to semantic similarity in a taxonomy. Lin achieves better results than distance-based definitions of similarity; his results correlate slightly better with human judgment than measures proposed by Resnik (1995) and by Wu and Palmer (1994). To illustrate the domain independence of his measure, Lin also applies it to the domain of ordinal values.

### 3 Approach

The Stanford Parser (de Marneffe et al., 2006) was applied to the Open American National Corpus (Ide and Suderman, 2004) to produce a database containing the counts of occurrences of all the dependency triples, which are of the form  $\langle \text{role}, \text{governor}, \text{dependent} \rangle$ , appearing in the corpus. Cover and Thomas (2006) define the information content of a proposition as the negative logarithm of its probability. We use this definition to compute the information content of the triples occurring in the corpus. Given a dependency triple, we define two predicates:

- A governor-position predicate substitutes a variable for the governor in the triple.
- A dependent-position predicate substitutes a variable for the dependent in the triple.

For example,

$t_1$ :  $\langle \text{dobj}, \text{grow}, \text{tomato} \rangle$

is one of the dependency triples occurring in the sentence:

$s_1$ : The gardener has grown tomatoes.

The governor-position predicate corresponding to  $t_1$  is:

$p_1$ :  $\langle \text{dobj}, \_G, \text{tomato} \rangle$

which binds to all occurrences of "tomato" as a direct object; the dependent-position predicate is:

$p_2$ :  $\langle \text{dobj}, \text{grow}, \_D \rangle$



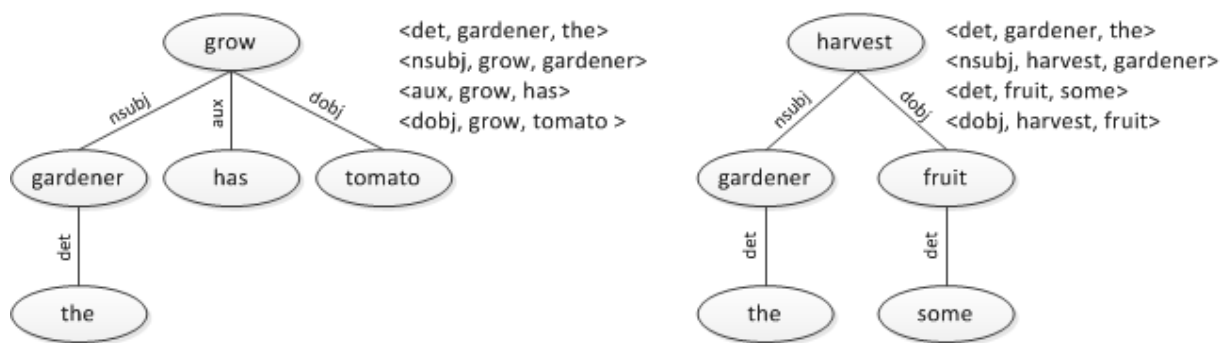


Figure 1: Dependency trees and dependency triples for  $s_1$  and  $s_2$

which binds to all occurrences of “grow” as a transitive verb. The information content of  $t_1$  is computed from the number of occurrences of instantiations of its dependent-position predicate. In general, let  $A$  be the number of occurrences of  $\langle r, g, d \rangle$  and let  $B$  be the number of occurrences of instantiations of  $\langle r, g, \_D \rangle$ . The information content of  $\langle r, g, d \rangle$  is defined by:

$$IC(\langle r, g, d \rangle) = -\log \frac{A}{B}$$

Next, we define similarity of two dependency triples using Lin’s information-theoretic definition of similarity. The definition is explained by an example computing the similarity between the following:

- $t_1$ :  $\langle \text{dobj}, \text{grow}, \text{tomato} \rangle$   
 $t_2$ :  $\langle \text{dobj}, \text{harvest}, \text{fruit} \rangle$

where  $t_2$  is a triple from the sentence:

$s_2$ : The gardener harvested some fruit.

The predicates  $p_1$  and  $p_2$  (above) are formed from  $t_1$ ; from  $t_2$ , we form the predicates:

- $p_3$ :  $\langle \text{dobj}, \_G, \text{fruit} \rangle$   
 $p_4$ :  $\langle \text{dobj}, \text{harvest}, \_D \rangle$

For each of these of these predicates, we form the set of all instantiations,  $M(p_n)$ . The numbers following the triples are hypothetical values for  $IC(t_n)$ :

- $M(p_1)$ :  $\{ \langle \text{dobj}, \text{grow}, \text{tomato} \rangle 1.7, \langle \text{dobj}, \text{raise}, \text{tomato} \rangle 3.8, \langle \text{dobj}, \text{eat}, \text{tomato} \rangle 2.4 \}$   
 $M(p_2)$ :  $\{ \langle \text{dobj}, \text{grow}, \text{tomato} \rangle 1.7, \langle \text{dobj}, \text{grow}, \text{strawberry} \rangle 2.7, \langle \text{dobj}, \text{grow}, \text{beard} \rangle 5.6 \}$

- $M(p_3)$ :  $\{ \langle \text{dobj}, \text{grow}, \text{fruit} \rangle 3.9, \langle \text{dobj}, \text{harvest}, \text{fruit} \rangle 7.2, \langle \text{dobj}, \text{eat}, \text{fruit} \rangle 1.2 \}$

- $M(p_4)$ :  $\{ \langle \text{dobj}, \text{harvest}, \text{tomato} \rangle 8.7, \langle \text{dobj}, \text{harvest}, \text{strawberry} \rangle 9.7, \langle \text{dobj}, \text{harvest}, \text{fruit} \rangle 7.2 \}$

For the two governor-position predicates,  $p_1$  and  $p_3$ , we compute the quotient of (1) the sum of the ICs of triples in  $M(p_1)$  and  $M(p_3)$  that have the same word in the governor position and (2) the sum of the ICs of all the triples in  $M(p_1)$  and  $M(p_3)$ . Triples that appear in both models are counted both times. Call this quotient  $S_g$ .

$$S_g = \frac{1.7 + 2.4 + 3.9 + 1.2}{1.7 + 3.8 + 2.4 + 3.9 + 7.2 + 1.2}$$

We form the quotient  $S_d$  similarly, using the dependent-position predicates. Finally, we define

$$\text{sim}(t_1, t_2) = \alpha \cdot S_g + (1 - \alpha) \cdot S_d$$

where  $\alpha$  is a real value between zero and one.

We extend this definition of similarity between triples to define similarity between sentences. Given two sentences, the nodes of their respective dependency trees are words and the tree edges are dependency relations. For example, the triple  $\langle \text{dobj}, \text{grow}, \text{tomato} \rangle$  indicates that *grow* and *tomato* are two nodes in the dependency tree and that there is a directed edge from *grow* to *tomato* labeled *dobj*.

Given two dependency trees and two nodes, one from each of the given trees, we form a collection of pairs where the first component is a branch that has the first node in the governor position and the second component is a branch that has the second node in the governor position. The process for forming this collection is as follows:

	$s_1$	$s_2$	Survey Average	Tree Similarity
1	The cat killed the mouse.	The mouse killed the cat.	0.5	0.633
2	The man walked to the store.	The person went to the store.	3.625	0.512
3	The student killed time.	The student killed the roach.	0.35	0.134
4	The janitor cleaned the desk.	The desk was cleaned by the janitor.	4.85	0.001
5	The locksmith went to the movies.	The window was stuck shut.	0.075	0.108
6	The dog went missing for three days.	The squirrel avoided the trap.	0.075	0.131
7	The student ran out of notebook paper.	The printer ran out of paper.	1.2	0.632
8	The door is open.	The door is closed.	0.5	0.330
9	Traffic downtown is heavy.	The downtown area is crowded.	2.4	0.075
10	The secretary stopped for coffee on the way to the office.	The office worker went out for dinner after work.	0.675	0.030
11	Biologists discovered a new species of ant.	Physicists verified the existence of black holes.	0.45	0.060
12	The artist drew a picture of the landscape.	The artist sketched a picture of the landscape.	4.375	0.675
13	The bear searched for food at the picnic grounds.	The bear scavenged the park for food.	3.525	0.500
14	A college degree allows one to have a rewarding career.	A bachelor's degree is necessary to get a high paying job.	2.125	0.294
15	The train arrives at half past three.	The visitor will be in the station this afternoon.	1.125	0.093

Table 1: Sentence Pairs with human subject survey averages and tree similarity measures. Survey averages range from 0 to 5; tree similarity measures range from 0 to 1.

- The triple with the highest information content from the collection of triples that have one of the given nodes in the governor position is identified. This triple may come from either tree.
- A search is done for the most similar triple from the other dependency tree.
- The two triples just identified are matched and removed from consideration. The process repeats until all of the branches exiting from one of the nodes have been matched.

Matching triples enables the recursive comparison of nodes from different dependency trees. We define the similarity of two nodes as the weighted average of:

- the similarity of the triples matched as described above;
- the result of recursively computing similarity of matched dependents (nodes one level

deeper in the dependency tree); and

- unmatched branches, defined as having a similarity of zero (The two nodes may have unequal numbers of children).

The similarity of two sentences is the similarity of their root nodes.

#### 4 Results

We applied the algorithm to 15 pairs of sentences written for the purpose of testing the approach. We asked 40 native English speakers to rank the similarity of each pair on a scale of 0 to 5, where 0 indicates “no overlap in meaning” and 5 indicates “complete overlap in meaning.” The tree similarity algorithm was applied to the sentence pairs. Table 1 shows the results (survey averages range from 0 to 5; tree similarity measures range from 0 to 1).

The two similarity measures have a correlation coefficient of 0.279; however, inter-annotator

agreement was low (Fleiss's kappa = 0.313). Pairs 7, 10, 14, and 15 had the lowest inter-annotator agreement. Without these pairs, the 11 pairs that remaine (1, 2, 3, 4, 5, 6, 8, 9, 11, 12, and 13) have kappa = 0.399 and have a correlation coefficient of 0.291 with the tree similarity algorithm. Pair 4, the active/passive switch, is incorrectly scored 0 by the algorithm, whereas the annotators were in strong agreement of a rating close to 5. Removing pair 4 from the analysis (which lowers kappa) gives a correlation coefficient of 0.618 between annotator averages and the algorithm results. These results suggest that, once the algorithm is refined to properly handle the active/passive switch, it will provide results that correlate to the judgment of native speakers.

## 5 Contributions & Future Work

Our approach is grounded in information theory. The representation avoids high-dimensional, sparse vectors; this allows the use of the trained database without having to condense it.

Previously Lang (2010) proposed implementing Lévi-Strauss's procedure for finding the structure of a myth (Lévi-Strauss, 1955). We plan to apply the tree similarity metric in a clustering algorithm for grouping sentences into categories corresponding to the constituent terms of his canonical formula.

## References

- Curt Burgess, Kay Livesay, and Kevin Lund. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25(2–3):211–257.
- Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. John Wiley & Sons, Hoboken, New Jersey, second edition.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 449–454, Genoa, May.
- Nancy Ide and Keith Suderman. 2004. The American National Corpus first release. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1681–1684, Lisbon, May.
- Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):10:1–10:25, July.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan, July.
- Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3):259–284.
- R. Raymond Lang and John G. Mersch. 2012. An experiment to determine whether clustering will reveal mythemes. In Mark A. Finlayson, editor, *Proceedings of the Third Workshop on Computational Models of Narrative*, pages 20–21, Istanbul, May.
- R. Raymond Lang. 2010. Considerations in representing myths, legends, and folktales. In *Computational Models of Narrative: Papers from the AAAI Fall Symposium*, pages 29–30, Arlington, VA, November.
- Claude Lévi-Strauss. 1955. The structural study of myth. *The Journal of American Folklore*, 68(270):428–444.
- Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, August.
- Dekang Lin. 1998. An information-theoretic definition of similarity. In Jude Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*, pages 296–304, Madison, Wisconsin, July. Morgan Kaufmann Publishers.
- J. L. McClelland and A. H. Kawamoto. 1986. Mechanisms of sentence processing: Assigning roles to constituents. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 2: Psychological and Biological Models*, pages 272–325. MIT Press, Cambridge, MA.
- Charles T. Meadow, Donald H. Kraft, and Bert R. Boyce. 1999. *Text Information Retrieval Systems*. Academic Press, Orlando, FL, 2nd edition.
- Naoaki Okazaki, Yutaka Matsuo, Naohiro Matsumura, and Mitsuru Ishizuka. 2003. Sentence extraction by spreading activation through sentence similarity. *IE-ICE Transactions on Information and Systems*, E86-D(9):1686–1694.
- Philip Resnik. 1995. Disambiguating noun groupings with respect to WordNet senses. In David Yarowsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 54–68, Cambridge, Massachusetts, June.
- Gerard Salton. 1988. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Roman Taraban and James L. McClelland. 1988. Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectations. *Journal of Memory and Language*, 27(6):597–632, December.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 133–138, Las Cruces, New Mexico, June. Association for Computational Linguistics.

# Effective Selectional Restrictions for Unsupervised Relation Extraction

Alan Akbik   Larysa Visengeriyeva   Johannes Kirschnick   Alexander Löser

Technische Universität Berlin

Databases and Information Systems Group

Einsteinufer 17, 10587 Berlin, Germany

firstname.lastname@tu-berlin.de

## Abstract

Unsupervised Relation Extraction (URE) methods automatically discover semantic relations in text corpora of unknown content and extract for each discovered relation a set of relation instances. Due to the sparsity of the feature space, URE is vulnerable to ambiguities and underspecification in patterns. In this paper, we propose to increase the discriminative power of patterns in URE using selectional restrictions (SR). We propose a method that utilizes a Web-derived soft clustering of  $n$ -grams to model selectional restrictions in the open domain. We comparatively evaluate our method against a baseline without SR, a setup in which standard 7-class Named Entity types are used as SR and a setup that models SR using a fine-grained entity type system. Our results indicate that modeling SR into patterns significantly improves the ability of URE to discover relations and enables the discovery of more fine-granular relations.

## 1 Introduction

In traditional approaches for Relation Extraction (RE), all target relations (such as BORNIN or HASWONPRIZE) need to be specified in advance. For each relation, an extractor is trained or manually created that finds relation instances in text (Jiang and Zhai, 2007). This process is expensive and usually involves manually labeling large amounts of training data, making it difficult to scale RE to large sets of relations. Worse, because target relations must be manually defined in advance, the usefulness of RE in corpora of unknown content is limited (Akbik et al., 2012). This limits their applicability to the open domain where a potentially unbounded number of relations may be expressed in text.

In contrast, Unsupervised Relation Extraction (URE) approaches do not require target relations to be pre-specified, and require no labeled training data (Rosenfeld and Feldman, 2007). Instead, they automatically *discover* prominent relations in a given text corpus and *extract* for each identified relation a list of relation instances. Current methods (Akbik et al., 2012; Yao et al., 2012) utilize a vector space model of semantics in which they group co-occurring pairs of entities (referred to as *entity pairs*) into clusters based on distributional evidence over observed patterns. Each cluster is interpreted as one discovered semantic relation and the entity pairs in each cluster as the instances of this relation.

**Pattern ambiguities.** However, a problem for such approaches is that patterns may be ambiguous in the sense that they point to more than one relation. The pattern “[X] GET [Y]”<sup>1</sup> for example may be observed for a person and a product (“*Jim got a new VW Beetle.*”), a person and a disease (“*Jim got H1N1.*”) or (colloquially) between a person and a difficult-to-understand topic (“*Jim finally got Game Theory.*”). Such ambiguous patterns can cause entity pairs that belong to different relations (such as <Jim, VW Beetle> and <Jim, H1N1>) to be falsely grouped into the same semantic relation. See Table 1 for a structured illustration of this example.

This is especially problematic because the number of observed patterns for each individual entity pair is usually disproportionately small compared to the space of all possible patterns. In such a sparse feature space, false evidence caused by ambiguities can potentially have a negative impact on

---

<sup>1</sup>Patterns are denoted with the placeholders [X] for the *subject* entity, and [Y] for the *object* entity of the entity pair they are observed with. In this paper, we use lexico-syntactic patterns extracted from dependency trees. For readability reasons, we omit information on dependency links. [X] in this pattern is either a subject or an apposition to the word GET. Likewise, [Y] is its object.

Sentence	Entity pair	Pattern	Restricted Pattern
<i>Yesterday, Jim got a new VW Beetle.</i>	<Jim, VW Beetle>	[X] GET [Y]	[X:PERSON] GET [Y:PRODUCT]
<i>Jim got H1N1.</i>	<Jim, H1N1>	[X] GET [Y]	[X:PERSON] GET [Y:DISEASE]
<i>Jim finally got Game Theory.</i>	<Jim, Game Theory>	[X] GET [Y]	[X:PERSON] GET [Y:THEORY]

Table 1: Example of pattern generation from three sentences. Three entity pairs are observed that belong to different relations. For example <Jim, VW Beetle> may belong to a PERSONACQUIREPRODUCT relation and <Jim, H1N1> to a PERSONINFECTEDWITHDISEASE relation. Without selectional restrictions, however, the same pattern is observed for all entity pairs, giving false evidence that they share the same relation. With selectional restrictions, different patterns are correctly observed.

the overall relation extraction quality of a URE approach.

**Selectional restrictions in patterns.** One approach to this problem is to include information on selectional restrictions (SR) to the patterns to increase their discriminative power (Resnik, 1996). We could restrict the patterns to apply only to entities of certain semantic classes or types. So, instead of the pattern “[X] GET [Y]” for the above mentioned examples we might generate “[X:PERSON] GET [Y:PRODUCT]” for “*Jim got a new VW Beetle.*”, “[X:PERSON] GET [Y:DISEASE]” for “*Jim got H1N1.*” and so forth (see Table 1).

However, modeling selectional restrictions in URE is not trivial, as it is unclear what type system and what *granularity* of types are required. For example, the types of a standard NER tagger (PERSON, LOCATION, ORGANIZATION etc.) may be too coarse grained for the above example, not being able to distinguish between DISEASE and PRODUCT.

While more fine-grained NER taggers have recently been researched (Ling and Weld, 2012), it is unclear whether they can be applied to the open domain. Here, we may encounter a potentially unrestricted set of entities of arbitrary types and granularity that varies from corpus to corpus. Also, each entity may have different types depending on how the type hierarchy is modeled; the string “*VW Beetle*” for instance may refer to a car, a product or a brand.

**Contributions.** In this paper, we address these challenges and study effective and viable methods for modeling selectional restrictions for URE in the open domain. We evaluate and discuss modeling SR using Named Entity types from the Stanford NER tagger (Finkel et al., 2005) as well as fine-grained Named Entity classes derived from the YAGO knowledge base (Hoffart et al., 2011). In addition, we propose a novel method that over-

comes shortcomings of existing methods by leveraging a Web-derived clustering of  $n$ -grams to model restrictions in an unsupervised fashion. We evaluate all setups against an informed baseline (based on previous work by (Akbik et al., 2012; Rosenfeld and Feldman, 2007)) in which patterns are not restricted.

We observe in all experiments that selectional restrictions significantly improve URE. The best performing setups use fine-grained Named Entity classes and our proposed open domain method, yielding  $f$ -measure improvements of 28% and 15% respectively over the baseline. We inspect the clustering results and find that the choice of SR influences the granularity of discovered relations. Based on our findings, we identify limitations of SR and outline challenges for URE.

## 2 Previous Work

We review previous work in URE with regards to selectional restrictions, and introduce the phrasal clustering dataset we use in our proposed method.

**URE.** There are a number of canonical works that relate to URE; (Lin and Pantel, 2001) first used distributional evidence to measure the similarity of patterns to find paraphrases of patterns. (Turney, 2006) instead computed the similarity of pairs of nouns using patterns as features. Their goal was finding analogies in text. (Rosenfeld and Feldman, 2007) then used a clustering method on a similar vector space model to group pairs of entities into clusters that represent semantic relations.

More recent work has addressed the problem of ambiguous patterns in URE in different ways. Notably, (Akbik et al., 2012) have evaluated pattern generation methods using lexical, shallow and deep syntactic features. They found that the use of deep syntactic features reduces pattern ambiguity and dramatically increases overall relation extraction  $f$ -measure by 65%. However, they do not model selectional restrictions in their pattern gen-

eration step.

**Selectional Restrictions.** Other recent work has incorporated information from NER taggers into their feature set. (Mesquita et al., 2010) use a standard 4-class NER tagger, but do not individually evaluate its impact. (Yao et al., 2012) use a very rich feature set, including fine-grained Named Entity types and document topics, to first disambiguate each pattern individually and in a second step perform URE using disambiguated patterns. This approach is problematic for many corpora because it requires a massive redundancy of pattern observations for disambiguation. In their experiments, they handled only patterns that are seen more than 200 times in their corpus. For comparison, in the large data set that we use in this paper, only 9 out of over 36.000 patterns are observed more than 200 times.

**Phrasal Clustering.** Contrary to previous work we do not propose using a manually established type system for selectional restrictions. Rather, we use a clustering of more than 10 million distinct one-to-five-word-grams from the Google  $n$ -gram data set (Lin and Wu, 2009) computed by (Lin et al., 2010). Previous work has leveraged the latent semantic information given by phrasal cluster memberships of  $n$ -grams to solve tasks other than URE. For example, (Zhou et al., 2011) increase the performance of deep syntactic parsers with regard to long-range dependencies, and (Täckström et al., 2012) transfer linguistic structure using cross-lingual word clusters.

In this work, we interpret each phrasal cluster as an entity type and all  $n$ -grams assigned to a cluster as belonging to this type. We incorporate this into the pattern generation step of our URE method and use this information to model selectional restrictions. Thus, the type system is not manually specified, but rather induced without supervision from a large Web corpus, making it a natural fit for the open domain and URE.

### 3 Pattern Generation

Pattern generation is the phase in URE in which patterns are generated for each co-occurring entity pair in the observed corpus. Current techniques go through each sentence in the corpus individually and generate <entity pair, pattern, count> tuples. In the following, we present the architecture of our URE system (Section 3.1) and illustrate how we integrate different options for modeling SR into

the pattern generation process. We present options that use types from an NER tagger (Section 3.2), fine-grained entity types from the YAGO knowledge base (Section 3.3), as well as the proposed phrasal clustering method (Section 3.4).

#### 3.1 Baseline System

In our system, we use a pattern generation method that makes use of dependency parses. We implement the algorithm described in (Akbik et al., 2012). Here, patterns are generated as a sequence of typed dependencies and lemmas of tokens on the shortest path between two entities in a parse. In addition to the tokens on the shortest path (referred to as *core tokens*), additional tokens are collected from their vicinity in the dependency tree. The position of the entities are denoted by the placeholders “[X]” and “[Y]”. We further prune patterns using linguistically-informed filters, e.g. removing patterns that consist only of direct dependencies between subject and object. We give an example of pattern generation applied to a sentence in Figure 1.

**Similarity of entity pairs.** Using this technique, we generate a list of pattern-entity pair observation tuples, which we use to construct a pair pattern frequency matrix. Each row vector represents one distinct entity pair  $e_i$  and each column one distinct pattern  $p_j$ . The value of the matrix cell  $c_{ij}$  is the number of times that  $e_i$  occurs in the pattern  $p_j$ . This representation allows us to compute the similarity of two entity pairs by computing the cosine distance between their corresponding rows in the pair pattern matrix (Bullinaria and Levy, 2007). We compute the pairwise similarity for all entity pairs to generate a dissimilarity matrix and execute a clustering method on this matrix.

**Clustering.** In line with most previous work in URE (Rosenfeld and Feldman, 2007; Wang et al., 2011), we use a Hierarchical Agglomerative Clustering (HAC) approach with the *average linkage* scheme (Han et al., 2011). This approach iteratively merges the two closest entity pairs to compute a dendrogram of cluster merges.

The dendrogram is cut at a point given by the *cutting threshold* parameter, yielding a set of clusters. This parameter is usually estimated or determined through experimentation. A common method is to execute an exhaustive search over a subset of the parameter space (referred to as

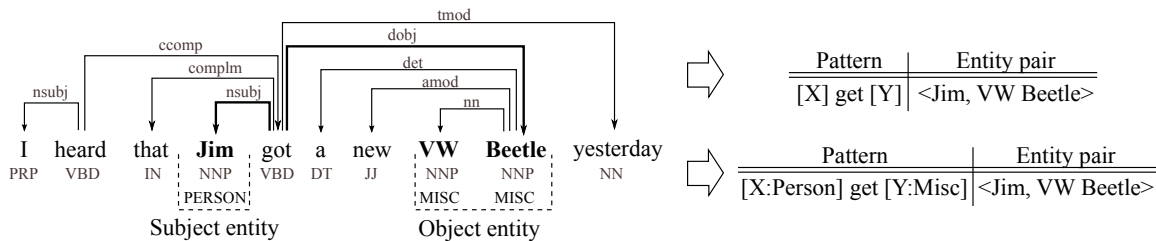


Figure 1: Illustration of the pattern generation process for one example sentence with the entities “Jim” and “VW Beetle”. Part-of-speech and Named Entity class tags are given below the tokens in the sentence. The shortest path between the two entities is highlighted bold in the dependency tree. The word “got” lies along this path, which is lemmatized to produce the pattern [X] GET [Y]. As an option, the Named Entity classes of the entities are included as selectional restrictions into the pattern, yielding the pattern [X:PERSON] GET [Y:MISC].

grid search), guided by cross-validation on a training set (Bergstra and Bengio, 2012). Through such experimentation, we determine that the cutting threshold must be set at a high value (for example around 0.999) to produce good clustering results<sup>2</sup>.

**Clustering result.** The clustering produces a set of clusters, which each consist of a set of entity pairs. Each resulting cluster is interpreted as one discovered relation, and all entity pairs in the cluster as the instances of this relation. The clustering result is then passed to an evaluation step discussed in detail in Section 4.

### 3.2 Named Entity Type Restrictions

We first extend the system with an option to use standard Named Entity types as selectional restrictions, in a similar fashion as a previous URE system (Mesquita et al., 2010). We incorporate the Stanford NER 7-class tagger into the sentence parsing pipeline and determine the type of each entity. These types are used to restrict the generic placeholders [X] and [Y] in generated patterns with the types of the subject and object entities.

For the example sentence illustrated in Figure 1, the tagger determines the class PERSON for “Jim”, and MISC for “VW Beetle”. The latter class is used for all entities that cannot be assigned any of the named classes. We therefore generate the pattern “[X:PERSON] GET [Y:MISC]” in this example. Because we model entity type restrictions directly into the patterns, we increase the space of all possible patterns and make individual patterns

<sup>2</sup>We determine different values for other linkage schemes. For example, when using the *single-link* linkage scheme in HAC, we find a good estimation for the cutting threshold to be 0.9.

more discriminative.

However, as shown in the example in Section 1, the Named Entity classes given by a 7-class tagger are coarse grained and may not include the types necessary to disambiguate all patterns. Also, there is a risk that Named Entity taggers may determine the wrong type for an entity<sup>3</sup>. This could lead to false evidence that negatively impacts URE.

### 3.3 Fine-grained Entity Type Restrictions

Because classes from a standard 7-class NER tagger may be too coarse grained for URE, we next extend the system with the option of modeling fine-grained Named Entity classes. We choose an approach that requires entities to be disambiguated and linked to Wikipedia URIs. The YAGO knowledge base then enables us to retrieve fine-grained entity classes for disambiguated entities, such as their Wikipedia categories (of which we use only the head nouns as restrictions). Because many YAGO entities belong to more than one class, this method returns a set of classes for each entity. For example, the Wikipedia categories for “VW Beetle” are “TAXICAB VEHICLES”, “AUSTRIAN INVENTIONS”, “INDUSTRIAL DESIGNS”, “SUBCOMPACT CARS” and others.

For each entity pair, we retrieve two sets of entity classes (one for the subject and one for the object). We determine the Cartesian product over these two sets and create one distinct pattern with selectional restrictions for each combination. For the example sentence, this means that we generate a list of patterns, including “[X:PERSON] GET [Y:CAR]”, “[X:PERSON] GET [Y:VEHICLE]” and “[X:PERSON] GET [Y:INVENTION]”, each of

<sup>3</sup>(Finkel et al., 2005) report an overall *f*-measure of 87% on the CoNLL 2003 Named Entity Recognition dataset.



1) Lookup phrasal clusters for subject and object entity

Pattern	Entity pair
[X] get [Y]	<Jim, VW Beetle>

<i>N</i> -gram lookup	Cluster	Weight	Other <i>n</i> -grams in cluster
VW Beetle	825	0.3	Chrysler Voyager, Toyota Highlander
VW Beetle	805	0.17	Computer Parts, Office Supplies
⋮	⋮	⋮	⋮
Jim	269	0.4	Becky, Doug, Eileen, Frances
⋮	⋮	⋮	⋮



2) Add cluster IDs as selectional restrictions to patterns.

Pattern	Entity pair	Weight
[X:269] get [Y:825]	<Jim, VW Beetle>	0.12
[X:269] get [Y:805]	<Jim, VW Beetle>	0.04
[X:283] get [Y:269]	<Jim, VW Beetle>	0.18
⋮	⋮	⋮

Figure 2: Illustration of the proposed pattern generation process that uses phrasal cluster memberships as selectional restrictions. In 1), phrasal clusters are retrieved for the subject and object of the entity pair. “VW Beetle” for example is in cluster 825, which contains many other car names. In 2) the Cartesian product over the phrasal clusters for subject and object is built and used as selectional restrictions for the pattern generated with the baseline method. This yields a set of patterns with different selectional restrictions.

which is used as a feature. While this method increases the overall number of observed patterns by about one order of magnitude, individual patterns are much more discriminative than without selectional restrictions.

**Limitations.** Two things must be noted regarding this method of determining fine-grained Named Entity classes. Firstly, it does not necessarily produce patterns at the desired granularity. In Section 1 we discussed the pattern “[X:PERSON] GET [Y:PRODUCT]” to be most appropriate, which is not generated by this method. More importantly though, the method is limited to entities that can be disambiguated to the appropriate Wikipedia page. While this is possible on the dataset we use for the evaluation, it is much more difficult to determine fine-grained Named Entity classes in the open domain with this method. We therefore implement this option mainly for evaluation purposes, in order to determine URE capabilities given a fine-grained, high quality type system for selectional restrictions.

### 3.4 Phrasal Clusters as Restrictions

To address the limitations of the methods described in 3.3, we propose a method for modeling SR that does not require an existing type system or the disambiguation of entities.

We extend the system with the option of using selectional restrictions derived from a phrasal

clustering computed by (Lin and Wu, 2009) over a dataset of more than 10 million distinct one-to-five-word-grams from the Google *n*-gram data set (Lin and Wu, 2009). In this dataset, each *n*-gram is assigned to ten different phrasal clusters with different association values, also referred to as *weights*. Weights are between 0 and 1, with a higher value indicating a stronger assignment confidence. Because the clustering is based on lexical context, *n*-grams in a cluster often share semantic properties. For example, the dataset contains clusters of entities like cities, cars, movies, etc (Lin and Wu, 2009).

During pattern generation, we look up the phrasal cluster IDs for the lexical representation of an entity and use this ID as selectional restriction. For example, the string “VW Beetle” belongs to phrasal cluster number 825 with weight 0.3. Semantically similar strings, such as “Chrysler Voyager” and “Toyota Highlander” are also part of this cluster. We can use this information to restrict the subject of the pattern only to strings that belong to cluster 825. Another phrasal cluster for “VW Beetle” is cluster 805 (with a lower weight of 0.17), which consists of more general product terms such as “Computer Parts” and “Office Supplies”. “Jim” is found in cluster 269, which contains many person first names. See Figure 2 for an illustration of this example.

We build the Cartesian product over the two

sets of phrasal clusters retrieved for the subject and object of an entity pair. Because each entity (e.g. its lexical representation) has 10 soft cluster memberships, the Cartesian product of phrasal clusters for both entities of an entity pair yields a total of 100 distinct weighted phrasal cluster ID combinations, hereafter referred to as *restriction pairs*. The weight of each restriction pair is computed by building the product of the confidence weights of the respective entity-phrasal cluster assignments. Each restriction pair is encoded into its pattern by adding to the entity placeholders “[X]” and “[Y]” a qualifier indicating the phrasal cluster ID. For each observation and restriction pair, a distinct pattern is generated.

This option increases the overall number of distinct patterns by two orders of magnitude. Patterns are also less humanly readable than their counterparts that use coarse- or fine-grained Named Entity types. We use this feature space to evaluate the assumption that we can leverage distributional evidence over a large Web corpus to model selectional restrictions in URE without an existing type system.

## 4 Evaluation

In this section, we perform experiments to measure the impact of different options of modeling selectional restrictions in patterns for URE. We also qualitatively inspect clusters and patterns.

### 4.1 Experimental Setup

Our experiments are performed on a silver standard dataset of 200.000 sentences crawled from the Web and labeled using distant supervision (Mintz et al., 2009). The sentences contain 4500 distinct entity pairs that are part of the YAGO knowledge base<sup>4</sup>. This allows us to compare the results of URE against the YAGO knowledge base. We compute BCubed (Amigó et al., 2009) precision, recall and  $f$ -measure values, which are commonly used to extrinsically evaluate clustering results. We perform this evaluation on the following setups:

**BASELINE** In this setup, we establish the URE quality of the baseline system (see Section 3.1) without modeling selectional restrictions. The baseline is based on the system described in (Akbik et al., 2012).

<sup>4</sup>In (Akbik et al., 2012) we illustrate and evaluate the labeling procedure in detail.

**NER-7CLASS** This scenario simulates previous work by (Mesquita et al., 2010). We evaluate the impact of using a standard NER tagger to model selectional restrictions (see Section 3.2).

**NER-YAGO** In this setup, we evaluate the use of a high quality, fine-grained type system to model selectional restrictions. We retrieve fine-grained entity classes from Wikipedia categories as described in Section 3.3.

**PROPOSED-OPEN-1** This setup is a modification of the proposed method that makes use of phrasal clusters to model selectional restrictions in the open domain. Here, we only use the cluster with the top weight (instead of all 10) as restriction for an entity.

**PROPOSED-OPEN-5** Like PROPOSED-OPEN-1 this is a modification of the proposed method. Here, the top 5 clusters for each string are used as restrictions. We use this setup to assess the impact of using only the most likely portion of the full phrasal clusters data set.

**PROPOSED-OPEN-FULL** The proposed method making use of the full phrasal clusters data set.

In addition to varying the pattern generation method we also experiment with different cutting thresholds in the Hierarchical Agglomerative Clustering method. We use two cutting threshold parameters that were determined through experimentation (see Section 3.1), namely 0.9995 and 0.9999 (referred to as  $C_{0.9995}$  and  $C_{0.9999}$  respectively). We also perform a grid search over the parameter space to determine the best cutting threshold for each setup, which we refer to as  $C_{best}$ .

### 4.2 Quantitative Evaluation

Table 2 shows the results of the quantitative evaluation. At all cutting thresholds, we observe improvements in overall  $f$ -measure with all setups (except PROPOSED-OPEN-1) over the baseline. These results indicate the value of including selectional restrictions in the pattern generation step of a URE method. When comparing the different methods, we note that NER-YAGO and PROPOSED-OPEN-FULL perform best, outperforming the baseline at peak setting by 15% and 28% respectively. PROPOSED-OPEN-1 performs much worse than the baseline, especially

	$C_{0.9995}$			$C_{0.9999}$			$C_{best}$		
	P	R	F1	P	R	F1	P	R	F1
BASELINE	0.34	0.59	0.43	0.21	0.74	0.33	0.46	0.45	0.46
NER-7CLASS	0.39	0.55	0.46	0.52	0.45	0.48	0.51	0.47	0.49
NER-YAGO	0.74	0.39	<b>0.52</b>	0.65	0.50	<b>0.57</b>	0.65	0.53	<b>0.59</b>
PROPOSED-OPEN-1	0.95	0.02	0.04	0.95	0.02	0.04	0.95	0.02	0.04
PROPOSED-OPEN-5	0.70	0.31	0.43	0.59	0.45	<b>0.51</b>	0.57	0.49	0.53
PROPOSED-OPEN-FULL	0.58	0.45	<b>0.51</b>	0.49	0.54	<b>0.51</b>	0.57	0.49	<b>0.53</b>

Table 2: Overview of the results of the comparative evaluation. At all cutting threshold settings, setups NER-YAGO and PROPOSED-OPEN-FULL achieve significantly higher  $f$ -measure scores than the baseline. We find that at peak performance, the PROPOSED-OPEN-5 setup reaches a similar quality as PROPOSED-OPEN-FULL.

BASELINE		
ID	Example patterns	Example entity pairs
1	[Y] OWNED BY [X], [Y] PART OF [X],	[X] BUY [Y], [Y] ACQUIRED BY [X] <SNCF, Systra> <Eskom, Arnet Power Station>
2	[X] WIN [Y], [Y] WINNING [X],	[X] RECEIVE [Y], [X] NOMINATED FOR [Y] <Cher, Emmy Award> <Chile, Chilean War for Independence>
3	[X] 'S SON [Y], [X] FATHER OF [Y]	[Y] BORN TO [X], [X] DAUGHTER OF [Y] <Zeus, Heracles> <Carus, Carinus>
4	[X] CREATE [Y], [Y] INVENTED BY [X],	[Y] BY [X], [Y] DEVELOPED BY [X] <Philipps, Compact Disc> <Kent Beck, Extreme Programming>
NER-YAGO		
ID	Example patterns	Example entity pairs
5	[X:FORMATIONS] FIGHT IN [X:WARS], [X:ORGANIZATIONS] WIN [Y:WARS], [Y:CONFLICTS] BETWEEN [X:ORGANIZATIONS]	<Red Army, Russian Civil War> <Rebel Alliance, Galactic Civil War>
6	[Y:PEOPLE] STUDENT OF [X:PHILOSOPHERS], [Y:PEOPLE] INFLUENCED BY [X:PHILOSOPHERS], [X:PHILOSOPHERS] TEACHER OF [Y:PEOPLE]	<Aristotle, Maimonides> <Ayn Rand, Ron Paul>
7	[Y:ALBUMS] BY [X:SINGERS], [Y:ALBUMS] ALBUM BY [X:MUSICIANS], [Y:ALBUMS] PERFORMED BY [X:MUSICIANS]	<Lou Reed, Coney Island Baby> <Bryan Adams, Reckless>
8	[Y:VENUES] HOME OF [X:TEAMS], [X:CLUBS] PLAY AT [Y:STADIUMS], [Y:CLUBS] AT [X:LOCATIONS]	<Milwaukee Brewers, Miller Park> <New York Yankees, Yankee Stadium>
PROPOSED-OPEN		
ID	Example patterns	Example entity pairs
9	[X:204] IN [Y:809], [X:809] IN [Y:764],	[X:204] IN [Y:764], [X:203] IN [Y:809] <Bob Gale, Back to the Future> <Cher, Zookeeper (film)>
10	[X:18] [Y:452] CANDIDATE, [X:793] [Y:284] CANDIDATE,	[X:233] [Y:441] POLITICIAN, [X:259] [Y:441] POLITICIAN <Bob Allen, Republican Party> <Fob James, Democratic Party>

Table 3: 10 sample clusters found with setups BASELINE, NER-YAGO and PROPOSED-OPEN. Each cluster is characterized by the top patterns in its centroid and represents one discovered relation. The entity pairs that make up the cluster are instances of discovered relations. Cluster 3, for example, represents the CHILDOF relation which holds between two persons.

with regards to recall. This is because this method produces highly overspecified patterns that do not allow for efficient grouping of entity pairs.

We also note that the cutting threshold setting has a significant impact on recall, precision and  $f$ -measure. At the PROPOSED-OPEN-5 setup, for example, minor variations in the parameter (from  $C_{0.9995}$  to  $C_{0.9999}$ ) cause an absolute  $f$ -measure difference of 0.8 points. This observation strongly indicates the importance of finding methods to ef-

fectively parameterize URE.

### 4.3 Qualitative Evaluation

We manually inspect a sample of the discovered relations and patterns to gain insight into how the different setups affect the relation discovery capabilities of our URE method. We illustrate our observations with a number of clusters shown in Table 3. We give examples of clusters for three setups: BASELINE, NER-YAGO and PROPOSED.

For each cluster, which represents one discovered relation, we list a small set of representative patterns and entity pairs.

Cluster 1, for example, is a cluster that represents company acquisitions, as is indicated by top patterns such as “[Y] ACQUIRED BY [X]” and “[Y] PART OF [X]”. Entity pairs in this cluster are relation instances. This means that <Eskom, Arnet Power Station> is an instance of the COMPANYACQUISITION relation. We find that this cluster corresponds most closely to the OWNS relation from the YAGO knowledge base.

**Readability.** Generally, we note that using named classes for selectional restrictions (NER-7CLASS and especially NER-YAGO) result in more human readable patterns than their counterparts in the baseline and proposed methods. Consider clusters 6 and 9. Cluster 6 is easy to evaluate, as the top patterns are human readable. It represents the INFLUENCEDBY relation that holds between a person and a philosopher. Cluster 9, on the other hand, is characterized by patterns that consist only of prepositions and phrasal cluster IDs. We must consult the entity pairs to determine that this cluster represents the ACTEDIN relation that holds between an actor and a film.

**Granularity.** In many cases, we find that selectional restrictions lead to the discovery of more fine-grained relations. An example of this is cluster 7, which denotes a relationship between a singer and the music album she created. All 46 instances in this cluster belong to the more general CREATED relation from YAGO that holds between a person and something she created (such as films, novels, albums etc.). This observation has implications for the use of selectional restrictions in URE, namely that the granularity of discovered relations can be influenced by the choice of type system. This also points to difficulties for the method of evaluating URE against an existing knowledge base as discovered relations might differ in granularity from the KB schema. Both these observations merit further investigation in future work.

**Ambiguities.** We look into errors made by the URE method and find that many errors are due to pattern ambiguities. Cluster 2, for example, mostly corresponds to the YAGO relation HASWONPRIZE. However, the patterns “[X] WIN [Y]” and “[Y] WINNING [X]” that hold between correct instances such as <Cher, Emmy Award>

also hold between false positives such as <Chile, Chilean War for Independence>. In more discriminative setups, this error is not made. For example, cluster 5 contains the more differentiated pattern “[X:ORGANIZATIONS] WIN [Y:WARS]”.

## 5 Conclusion and Outlook

In this paper, we addressed the problem of pattern ambiguities in URE by evaluating different methods of modeling selectional restrictions. We find that SR generally have a positive impact on relation discovery capabilities of our URE method. Significantly, we find a fine-grained type system to be the best setting, especially if URE is applied to a closed domain where most types of interest can be detected. For the open domain, we have presented a method that makes use of a Web-derived phrasal clustering of  $n$ -grams. We find our proposed method to be effective in reducing pattern ambiguities, with the advantage of being independent of a manually determined type system. Based on our results, we believe that correctly restricted deep syntactic patterns are the best features for URE.

In a qualitative evaluation of clustering results, we have determined two main issues that merit being addressed in future work in URE. First, automatic evaluation of URE remains problematic, as relations might be discovered that differ in granularity or semantics from the knowledge base that is evaluated against. Current evaluation methods penalize such divergence, even though discovered relations might still be correct. Second, we found that the parameterization of the clustering approach used in URE greatly influences the result quality and granularity. We find that even minor variations on the cutting threshold parameter for Hierarchical Agglomerative Clustering greatly impact overall  $f$ -measure.

Future work will focus on closely investigating clustering techniques and methods for effective parametrization. In addition, we intend to investigate Active Learning (Sun and Grishman, 2012) as a method to include minimal amounts of human feedback to guide the relation discovery process and improve overall URE results.

## Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Alan Akbik received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT-2009-4-1 270137 ‘Scal-

able Preservation Environments’ (SCAPE). Larysa Visengeriyeva, Johannes Kirschnick and Alexander Löser received funding from the Federal Ministry of Economics and Technology (BMWi) under grant agreement “01MD11014A, ‘MIA-Marktplatz für Informationen und Analysen’ (MIA)”.

## References

- A. Akbik, L. Visengeriyeva, P. Herger, H. Hemsén, and A. Löser. 2012. Unsupervised discovery of relations and discriminative extraction patterns. In *Proceedings of the 24th International Conference on Computational Linguistics*.
- Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, August.
- James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305.
- J.A. Bullinaria and J.P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.
- 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*, pages 363–370. Association for Computational Linguistics.
- Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
- Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, WWW ’11, pages 229–232, New York, NY, USA. ACM.
- Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120.
- Dekang Lin and Patrick Pantel. 2001. Dirt: discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.
- D. Lin and X. Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL ’09, pages 1030–1038, Stroudsburg, PA, USA. Association for Computational Linguistics.
- D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, et al. 2010. New tools for web-scale n-grams. In *Proceedings of LREC*.
- Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.
- F. Mesquita, Y. Merhav, and D. Barbosa. 2010. Extracting information networks from the blogosphere: State-of-the-art and challenges. In *Fourth Int’l AAAI conference on weblogs and social media*.
- M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL/AFNLP*, pages 1003–1011.
- P. Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1):127–159.
- B. Rosenfeld and R. Feldman. 2007. Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 411–418. ACM.
- Ang Sun and Ralph Grishman. 2012. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1105–1112. ACM.
- O. Täckström, R. McDonald, and J. Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. *Proceedings of the North American Chapter of the Association for Computational Linguistics*.
- P.D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.
- W. Wang, R. Besançon, O. Ferret, and B. Grau. 2011. Filtering and clustering relations for unsupervised information extraction in open domain. In C. Macdonald, I. Ounis, and I. Ruthven, editors, *CIKM*, pages 1405–1414. ACM.
- Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 712–720. Association for Computational Linguistics.
- G. Zhou, J. Zhao, K. Liu, and L. Cai. 2011. Exploiting web-derived selectional preference to improve statistical dependency parsing. In *Proceedings of ACL*, pages 1556–1565.

# Bootstrapping Semantic Lexicons for Technical Domains

Patrick Ziering<sup>1</sup> Lonneke van der Plas<sup>1</sup> Hinrich Schütze<sup>2</sup>

<sup>1</sup>Institute for NLP, University of Stuttgart, Germany

<sup>2</sup>CIS, University of Munich, Germany

{Patrick.Ziering, Lonneke.vanderPlas}@ims.uni-stuttgart.de

## Abstract

We address the task of bootstrapping a semantic lexicon from a list of seed terms and a large corpus. By restricting to a small subset of semantically strong patterns, i.e., coordinations, we improve results significantly. We show that the restriction to coordinations has several additional benefits, such as improved extraction of multiword expressions, and the possibility to scale up previous efforts.

## 1 Introduction

High-quality semantic lexicons are needed for many natural language processing (NLP) tasks like information extraction and discourse processing (Riloff and Shepherd, 1997). Building such lexicons manually is costly and time-consuming. Automatic lexicon construction is therefore an important task and much prior work has addressed it.

This paper adopts *Basilisk* (Thelen and Riloff, 2002) as its basic approach, a system that uses lexico-syntactic patterns for bootstrapping. We have adapted *Basilisk* to our setting in several ways. Whereas the original *Basilisk* covers a wide variety of lexico-syntactic patterns, we restrict ourselves to a specific type of patterns, i.e., coordinations. Coordinations have been exploited in lexical acquisition before (e.g., Roark and Charniak (1998); Caraballo (1999); Goyal et al. (2010)). Most of this previous work uses a pairwise perspective (i.e., a focus on whether two words co-occur in a coordination). However, we use coordinations in a *Basilisk* approach, for which patterns contain several terms in general. We therefore do not split up coordinations in pairs but keep the complete coordination intact. Coordinations in technical domains frequently contain more than 2 elements.

We argue that bootstrapping methods, known to be particularly sensitive to the ambiguity of terms

and contexts and prone to semantic drift, benefit from the strong semantic coherence found in coordinations. The elements of a coordination often have a common hypernym; i.e., they are co-hyponyms or members of a common semantic class.

Furthermore, the high arity of coordinations, the fact that coordinations have two or more arguments (e.g., “platinum, nickel, palladium, copper, silver, or gold”) further constrains the semantics. For example, if two out of three elements in a coordination have already been identified as substances, this makes it likely that the third is also a substance. General or lexico-syntactic patterns in lexical bootstrapping have arity 1, i.e., they have a single argument. These patterns are too often not restrictive enough to prevent false terms infecting the semantic lexicon, which can lead to semantic drift. For example, given a corpus in which the semantic class DISEASE is not predominant, after some iterations the weak pattern “treatment of <X>” is selected, which also provides many off-class candidates (e.g., treatment of *prisoners*). In coordinations, the semantic coherence among terms leads to more selective patterns. For example, the coordination “congenital heart defect, atherosclerosis, <X>, scleroderma or tuberos sclerososis” can only hold a slot for diseases.

We will show that the restriction to coordination patterns has several additional benefits. The focus on simple coordination patterns circumvents the need for identifying various syntactic relations (e.g., subject), that are part of the extraction patterns of the original *Basilisk*. Therefore, it circumvents the need for parsing. Syntax in the patent domain, we address in this paper, is complex and characterized by long clauses (cf. average number of tokens per sentence: BNC: 19.7; Brown: 21.3; WSJ: 22.4; Wikipedia: 24.3; EPO: 32.4). The shallow parser used by Thelen and Riloff (2002) would need several months to parse our corpus.

Furthermore, the restriction to a subset of very strong bootstrapping patterns limits the overproduction of patterns radically. We can therefore apply our method to the largest domain-specific corpus that has been used for semantic bootstrapping so far.

Lastly, we benefit from the increased precision in extracting multiword expressions (MWEs) when using coordination patterns. In contrast to original Basilisk and most prior work, we learn terms of any length, because, as we will see later, in the technical domain under consideration MWEs are predominant (e.g., “alkyl trimethyl ammonium methosulfate”).

The paper is structured as follows. In Section 2, we describe our data set, the task we address and our evaluation methodology. Section 3 describes Basilisk and our adaptations, in particular, the context patterns we use. Experimental setup and results are presented in Section 4. In Section 5, we discuss and analyze these results. The last two sections describe related work and conclusions, respectively.

## 2 Data, task description and evaluation methodology

**Data.** We use the patent data distributed by the European Patent Office<sup>1</sup> (EPO) as our corpus. We extract the description (the main part of a patent) from 561,676 English patents filed between 1998 and 2008 and perform sentence splitting and tokenization using Treetagger (Schmid, 1994) and lemmatization and part-of-speech (POS) tagging using MATE (Bohnet, 2010). Sentences up to a size of 100 tokens extracted from a sample of 25,000 patents are parsed by MATE. The resulting EPO corpus consists of roughly 4.6 billion tokens.

**Task description.** The task we address is semantic tagging of patents. The research reported here was conducted as part of a project on computational linguistics analysis of patent text. We want to be able to support functionalities like color-coding entities of a particular semantic class for quick perusal; or searching for entities in a particular semantic class. Our longterm goal is to support semantic tagging for a large variety of semantic classes. In this paper, we focus on the semantic classes SUBSTANCE and DISEASE. A substance is a particular kind of physical matter with uniform properties. Substances are of obvious relevance

for the patent domain and a large proportion of patents contain substances. A disease is an abnormal condition that affects the body of an organism. We selected disease as a clearly nontechnical category to be able to investigate potential differences of lexical bootstrapping algorithms for categories with very different properties.

Gazetteers are crucial for good performance in machine-learning-based semantic tagging (Ratinov and Roth, 2009), e.g., the best performing systems for recognition of person, location and organization named entities all use gazetteer features (e.g., Florian et al. (2003)). It is in this context that we address the task of bootstrapping lexicons from corpora: for most semantic classes of interest in the patent domain high-coverage lexicons are not available.

**Evaluation methodology.** Since our primary task is semantic tagging, we evaluate the quality of the bootstrapped lexicon directly on this task, i.e., on the task of tagging members of the semantic class in text – rather than evaluating the lexicon in a type-based evaluation as a set of terms without context as most previous work has done. A tagging-based evaluation directly measures what we need for our application, e.g., frequent terms have a higher impact on tagging accuracy than rare terms, and ambiguous terms with a rare class sense depress tagging accuracy.

**Terminology.** From now on, we use the term *MWE* for a noun phrase that we identify as a candidate class instance; we include one-word noun phrases in the definition of MWE in this paper. We call an MWE in a particular context in our gold standard a *gold-standard MWE* if it was annotated as a member of the semantic class in question. We call an MWE a *lexicon MWE* if our bootstrapping algorithm has added it to the induced lexicon as a class instance.

**Gold standard creation.** Asking human annotators to mark all instances of SUBSTANCE/DISEASE in a randomly selected set of patents is very inefficient because this would result in annotators spending a lot of time reading patent text that contains (almost) no class instance. Moreover, annotation quality is higher in patents that contain at least a moderate number of class instances since annotators will remain alert as they go through the document.

To address this problem, for the two classes we stratify the EPO corpus into three strata according

<sup>1</sup>www.epo.org

to density  $\rho$ : high, medium and low.  $\rho$  is computed as the proportion of class instances per token. Since the low-density stratum contains virtually no class instances, we exclude it from our experiments.

We randomly select 1000 patents from each of the medium-density and high-density strata and then one sentence from each patent. One annotator labeled 200 sentences using the GATE<sup>2</sup> annotation tool. Then problematic annotation examples were discussed. Afterwards, the annotator labeled the remaining 1800 sentences. For assessing the quality of the gold standard, a second trained annotator labeled 200 sentences of our evaluation set. Inter-annotator agreement for both classes was  $\kappa = .712$  (macro kappa) and  $\kappa = .818$  (micro kappa) (Cohen, 1960), which indicates substantial to excellent agreement (Landis and Koch, 1977).

### 3 Bootstrapping algorithms

```

1: lexicon ← seed
2: for int  $i = 0; i < m; i++$  do
3:   patterns ← patternsOf(lexicon)
4:   score(patterns)
5:   patterns ← return-top-k(patterns, 20 +  $i$ )
6:   terms ← termsOf(patterns) – lexicon
7:   score(terms)
8:   lexicon ← lexicon  $\cup$  return-top-k(terms, 5)
9: end for
10: return lexicon

```

Figure 1: Basilisk algorithm. The original version of Basilisk defines terms as head nouns.

The basic bootstrapping algorithm we use is Basilisk as shown in Figure 1 (Thelen and Riloff, 2002). Basilisk first initializes the lexicon as the seed set (line 1). The basic idea of the algorithm is to identify context patterns that reliably identify lexicon terms (lines 3–4), e.g., *made of*  $\langle X \rangle$ . Line 5 selects a subset of patterns<sup>3</sup> based on the scoring function  $R\log F$ :

$$R\log F(\text{pattern}_i) = F_i/N_i \log_2(F_i)$$

where  $F_i$  is the number of learned lexicon terms that occur in  $\text{pattern}_i$  and  $N_i$  is the total number of terms occurring in  $\text{pattern}_i$ . Lines 6–7 select the terms associated with the patterns selected on line 5 and score them. Terms are scored using  $\text{AvgLog}$ , the average log frequency, (line 7):

<sup>2</sup>gate.ac.uk

<sup>3</sup>We discard patterns that only occur with already learned terms, guaranteeing that each of the selected  $20 + i$  patterns on line 5 can potentially contribute new terms

$\text{AvgLog}(\text{term}_i) = 1/P_i \sum_{j=1}^{F_i} \log_2(F_j + 1)$  where  $P_i$  is the number of patterns in which  $\text{term}_i$  occurs and  $F_j$  is the number of learned lexicon terms that occur in  $\text{pattern}_j$ . The 5 highest scoring terms are then added to the lexicon (line 8).

#### 3.1 Basilisk-G

To process the large-scale patent corpus efficiently, we implemented a simple chunker that identifies noun phrases (NPs) using regular expressions (REs) on POS tags. We refer to this RE/POS-based algorithm as *Basilisk-G* – for “Basilisk General Patterns”. Basilisk-G is an instantiation of the basic Basilisk algorithm in Figure 1 that uses a more general definition of patterns that only requires POS tagging and no parsing. We use all patterns of the form  $w_{-i} \dots w_{-1} \langle \text{np} \rangle w_1 \dots w_j$  where  $0 \leq i, j \leq 3$  and  $i + j \geq 2$ ; i.e., context extends up to three tokens out to the left and right from  $\langle \text{np} \rangle$  and must consist of at least two tokens. We discard patterns whose context does not contain a verb or a noun. The standard version of Basilisk uses pattern templates like  $\langle \text{subj} \rangle$  *verb* and *noun prep*  $\langle \text{np} \rangle$ . Our more general definition covers most instantiations of these templates, but it extends the patterns considered to a much larger variety of lexical contexts. For example, fragments of a coordination like *, silver, <np>* or *platinum* are also instantiations of the general Basilisk-G pattern template. As we will see later, these types of patterns (which original Basilisk does not use) turn out to be very effective.

Similar to patterns, we define MWEs in Basilisk-G (Figure 1, line 6) as part of an NP extracted using REs: an MWE is a (possibly zero-length) sequence of prehead modifiers (adjectives and nouns) terminated by the head.

#### 3.2 Basilisk-C

In this section we introduce *Basilisk-C* – for “Basilisk Coordination Patterns”, a Basilisk instantiation that uses only coordinations.

We allow two different types of coordinations: *and/or* coordinations and punctuation coordinations.

An *and/or* coordination is a list of NPs consisting of two parts. In the first part of the list, NPs are separated by commas or semicolons. In the second part, NPs are separated by “and”, “or” or “and/or”. The second part has minimum length 2:

$$((\text{NP}, )*(\text{NP}; )*) \text{NP} ((\text{and}/\text{or})?|\text{or}) \text{NP} +$$



A punctuation coordination is defined as a list of at least three NPs separated by commas or semicolons:  $((NP, )+ NP, NP) \mid ((NP; )+ NP; NP)$ <sup>4</sup>

Because we detect the coordinations and the NPs based on POS REs without performing a full syntactic analysis and because the assumption of co-hyponymy is incorrect in certain cases, there are several incorrect matches. We automatically removed border elements of a coordination if they indicate an extraneousness to the coordination. One indicator of extraneousness was the unique presence of a determiner for example “this description” in “as concrete examples of [this description, methyl alcohol and benzyl alcohol] may be cited”, where [...] matches the coordination pattern. Moreover, we removed conjuncts that indicate a hypernym relation such as “other products” in “copolymers, polyisobutene and other products”.

We treat the conjuncts that survive filtering as an unordered set, i.e., we ignore their order in the text. The set is discarded and not used by Basilisk-C if it only contains one element.

## 4 Experimental setup and results

**Experimental setup.** We evaluate performance of the two algorithms Basilisk-G and Basilisk-C introduced above. We run experiments on EPO (Section 2) with the goal of learning the classes SUBSTANCE and DISEASE. Our seed set (Figure 1, line 1) consists of the 4223 substances distributed by Ciaramita and Johnson (2003) as part of SuperSenseTagger and 239 diseases extracted from Simple English Wikipedia<sup>5</sup>.

For Basilisk-C, we extracted 9.7 million unique coordinations, out of a total of 25 million.

For Basilisk-G, we found 1.6 billion unique context patterns. In order to be able to run experiments quickly, we introduce frequency thresholds for MWEs, patterns and MWE-pattern combinations. We only consider MWEs and patterns that occur at least  $\theta_1 = 10$  times and MWE-pattern combinations that occur at least  $\theta_2 = 3$  times in EPO. These thresholds are unlikely to diminish lexicon quality since many rare instances of MWEs are due to OCR errors or failures of our RE-based recognition of NPs (see also (Qadir and Riloff, 2012)). Using the thresholds  $\theta_1$  and  $\theta_2$ ,

<sup>4</sup>This version of Basilisk uses the same RE to detect NPs as Basilisk-G.

<sup>5</sup>[simple.wikipedia.org/wiki/List\\_of\\_diseases](http://simple.wikipedia.org/wiki/List_of_diseases)

there were 3.2 million unique MWEs, 56 million unique patterns and 121 million unique MWE-pattern combinations. This is the raw data we run Basilisk-G on.

As discussed in Section 2, our evaluation methodology directly evaluates the semantic lexicon on the task of interest: semantic tagging of patents. The tagging method we use is simple lexicon lookup. While tagging MWEs we exploit the compositional structure of entities by merging adjacent or overlapping token-based labels (e.g., *fatty acid* and *acid amide* are merged to *fatty acid amide*). In our decision to use lexicon lookup for tagging, we follow Qadir and Riloff (2012), who argue convincingly that for a specialized class and domain, ambiguity of terms (which would be the main reason for using a context-dependent method like a CRF) is a limited phenomenon and ignoring it does not greatly affect performance. Even so, it is important to keep in mind that tagging precision does not directly reflect lexicon accuracy.

We use the measures precision, recall and  $F_1$ . Tagging results are evaluated using the evaluation module of GATE. The scores give half credits for partial matches and full credits for exact matches.

### Performance of Basilisk-G and Basilisk-C.

Table 1 shows the performance of the baseline and of Basilisk-G and Basilisk-C for different lexicon sizes. The baseline uses the seed set (SUBSTANCE: 4223; DISEASE: 239) for tagging. We first run iterations until the size of the induced lexicon is 5000 and then double the lexicon three times – to 10,000, 20,000 and 40,000 – to investigate the relationship between lexicon size and tagging performance.

Both Basilisk-G and Basilisk-C consistently beat the baseline in recall and  $F_1$  for DISEASE and in all three measures for SUBSTANCE. We mark each performance number with a star if it is significantly higher than the number above it.<sup>6</sup> For example, Basilisk-G’s and Basilisk-C’s  $F_1$  of .549 for 10,000 substances is significantly better than the baseline (.539).

Basilisk-C outperforms Basilisk-G in most cases. We mark each Basilisk-C performance number with † if it is significantly higher than the Basilisk-G number to the left of it. The largest differences between Basilisk-C and Basilisk-G can be found for the smaller semantic class of diseases and at larger lexicon sizes. This is to be expected

<sup>6</sup>Approximate randomization test (Yeh, 2000),  $p < .05$

size	SUBSTANCE						DISEASE					
	P		R		$F_1$		P		R		$F_1$	
	B-G	B-C	B-G	B-C	B-G	B-C	B-G	B-C	B-G	B-C	B-G	B-C
seed	.597		.491		.539		.793		.233		.360	
5000	.599*	.598	.494*	.492*	.542*	.540*	.790	.724	.455*	.556*†	.578*	.629*†
10,000	.605*	.604*	.502*	.504*	.549*	.549*	.476	.643†	.645*	.602*	.548	.622†
20,000	.610*	.614*	.509*	.529*†	.555*	.568*†	.392	.530†	.642	.701*†	.487	.604†
40,000	.612*	.619*	.515*	.549*†	.559*	.582*†	.300	.473†	.642	.720†	.409	.571†

Table 1: Tagging performance measured by precision (P), recall (R) and  $F_1$  of seed baseline and for different lexicon sizes of Basilisk-G (B-G) and Basilisk-C (B-C); \* indicates significantly higher than the number above it; † indicates significantly higher than the number to the left of it.

because we would expect semantic drift to be more prominent for smaller classes and to grow with the size of the induced lexicon. Closer inspection reveals that Basilisk-G indeed drifts to any kind of technical properties. For substances, Basilisk-C outperforms Basilisk-G mainly in recall. This large class is less sensitive to semantic drift but, as we will discuss in detail in the error analysis, still benefits from the MWE extraction of Basilisk-C. These results support the argument we have made for restricting to coordinations in Basilisk for both predominant classes such as SUBSTANCE and minor semantic classes such as DISEASE.

Note that the best performance for the small class of diseases is found at a lexicon size of 5000:  $F_1 = .629$ . It outperforms the seed baseline (+.269) and Basilisk-G (+.051). Precision drops rapidly when doubling the lexicon size and introducing more and more semantic drift. For the large semantic class of substances we find that increasing the lexicon size generally improves performance. The overall best result achieved,  $F_1 = .582$ , is achieved by Basilisk-C and the largest lexicon size (40,000).

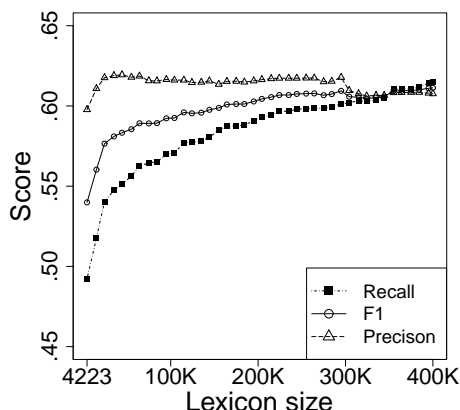


Figure 2: Performance of Basilisk-C as a function of lexicon size for substances

Our comparisons between Basilisk-G and Basilisk-C are limited to a lexicon size of 40,000 because running Basilisk-G for larger lexicons be-

comes infeasible in our current setup. However, one of the additional advantages of restricting to coordinations is scalability. Figure 2 shows the performance of Basilisk-C as a function of lexicon size for very large substance lexicons. The figure suggests that there is an upward trend for  $F_1$  and recall up to very large lexicon sizes. The curves do not increase monotonically, partly due to the fact that once the lexicon has reached a size of more than 100,000, only a few additional MWEs found in the evaluation corpus are responsible for the changes. Thus, the curves do not directly reflect actual expected performance. Basilisk-C achieves the highest performance at the right edge of the graph at lexicon size 400,000:  $F_1 = .611$ ,  $P = .608$ ,  $R = .615$ . To our knowledge, Basilisk-type algorithms have not been run for lexicons of this size before.

The tagging performance seems low for practical purposes. However, this reflects the lexicon quality (i.e., the bootstrapping performance) only partially. Since we are using a large domain-specific patent corpus, we are faced with many high-specific and infrequent terms. Many false positives arise because of class instances missed by the annotators. An optimal gold standard for this domain would require domain specialists.

**Comparison with original Basilisk.** We already explained that running original Basilisk on the whole EPO corpus is not feasible due to its size and the length of the sentences. However, for a comparison of Basilisk-G and Basilisk-C with the original Basilisk, we parsed sentences up to a size of 100 tokens from 25,000 sample patents. We define *Basilisk-LS* – for “Basilisk Lexico-Syntactic patterns” and extract all lexico-syntactic patterns described in (Riloff and Phillips, 2004). We introduce two versions of Basilisk-LS: the originally head-based *Basilisk-LS<sub>head</sub>* and a variant, *Basilisk-LS<sub>MWE</sub>*, that extracts MWEs defined as basic noun phrases without determiner or post-

nominal phrases such as PPs or relative clauses. From all parsed sentences, we extract POS-based coordinations and all general patterns for our systems. The tagging results for a substance lexicon of size 20K is shown in Table 2.

System	P	R	$F_1$
Basilisk-LS <sub>head</sub>	.437	.502	.467
Basilisk-LS <sub>MWE</sub>	.582	.506	.541
Basilisk-G	.560	.524	.542
Basilisk-C	.620	.567	.592

Table 2: Comparison of B-LS, B-G and B-C

The results for Basilisk-LS<sub>head</sub> show that precision is a severe problem in tagging substance head nouns on patents. When inducing and tagging MWEs as it has been done by the other systems, precision is much higher. Both Basilisk-LS<sub>MWE</sub> and Basilisk-G only slightly outperform the seed baseline in  $F_1$  (Table 1). Once more Basilisk-C outperforms all other systems by far.<sup>7</sup>

**Extension on other classes and domains.** To show that Basilisk-C can be applied to other classes and domains, we ran an additional experiment. We applied Basilisk-C to the class FIXED-LOCATION as defined by Qadir and Riloff (2012) and used Wikipedia coordinations. As there was no annotated data available for this class, we ran a type-based evaluation. A human judge rates accuracy of 100 samples of the lexicon. As seed set, we selected five US states, five European countries and ten national capitals. We induced 5000 fixed locations with an accuracy of 80%. We conclude that depending on corpus size lexical bootstrapping based on coordinations is applicable to any domain for several classes.

## 5 Analysis and discussion

**High arity.** In previous sections we explained that the high arity of the coordination relation constrains the semantics of its arguments and remedies problems related to ambiguity that can give rise to semantic drift. We have seen this in the results. The relatively small class (for our domain) of diseases is prone to semantic drift especially

<sup>7</sup>Recall of Basilisk-C for the subcorpus is even better than on the full EPO corpus shown in Table 1. The reason for this is that sentences up to 100 tokens contain shorter coordinations. MWEs in short coordinations tend to be less specific and thus more frequent in a test set. This explains why recall of semantic tagging is better when using shorter coordinations.

when larger lexicon sizes are induced. Basilisk-C is able to remedy this problem and leads to higher performances than Basilisk-G and the differences are largest for larger lexicon sizes.

On the other hand, we discussed the phenomenon that MWEs in short coordinations tend to be less specific. Despite the fact that shorter coordinations provide a smaller pool of term candidates, we expect recall in the semantic tagging task to be higher when using shorter coordinations because the available term candidates are less specific and thus have a higher frequency, i.e., a higher chance to occur in the test set. Table 3 shows the tagging performance of a lexicon with 20,000 substances and one with 10,000 diseases induced by Basilisk-C applied to different ranges of coordination lengths. It shows that Basilisk-C using only coordinations up to a size of 5 terms (“2 to 5”) outperforms Basilisk-C using all coordinations (“2 to  $\infty$ ”) in recall. In predominant classes such as SUBSTANCE, shorter coordinations do not harm precision. However, for classes like DISEASE, precision decreases when shorter coordinations are used, as illustrated in Table 3.

Coordination length	P	R	$F_1$
20,000 substances			
2 to $\infty$	.614	.529	.568
2 to 5	.615	.568	.591
10,000 diseases			
2 to $\infty$	.643	.602	.622
2 to 5	.470	.645	.544

Table 3: Comparison of coordination lengths

**High-confidence pattern.** We argued in subsection 3.2 that coordinations impose a stronger semantic coherence on MWEs than general context patterns or lexico-syntactic patterns do. Table 4 shows that coordinations are indeed high-confidence patterns for learning substances. High-confidence Basilisk-G patterns after 1000 and 6000 iterations are listed. Each of the top 20 patterns after 1000 iterations is a coordination. Apparently, the patterns that are selected in the beginning of learning as the ones best suited for identifying substances are all fragments of coordinations. Thus, performance of Basilisk-G and Basilisk-C for a lexicon of 10,000 MWEs (cf. Table 1) is fairly equal.

In contrast, after 6000 iterations only three of the highest-confidence patterns still are coordinations (not shown) – the other 17 are other types of

	type of pattern	example
$i = 1000$	NN , <np> ,	nitrate , <np> ,
	, <np> , NN	, <np> , magnesium
$i = 6000$	NN , <np> CC	oxide , <np> , and
	, <np> , NN	, <np> , sodium
	NN of <np> (	weight of <np> (
	of <np> verb	of <np> was added

Table 4: Highest-confidence Basilisk-G patterns after  $i$  iterations (examples from top 20)

patterns (Table 4,  $i = 6000$ ). As Basilisk-G’s performance improves more slowly than performance of Basilisk-C after the initial iterations (cf. Table 1, 20,000 to 40,000), we conclude that coordinations are the most effective patterns and the addition of other pattern types contribute little to learning new substances.

**Scalability.** Besides the scalability upgrade in preprocessing by avoiding parsing, Basilisk-C, in particular ranking of coordinations and MWEs, runs quicker as the input of term-pattern combinations is about 80% smaller than for original Basilisk. This means a crucial scalability benefit for corpora even larger than EPO.

**MWE extraction.** The results in Table 2 show that inducing and tagging only head nouns rather than MWEs leads to poor precision. This result is to be expected as our set of gold-standard MWEs comprises 45.4% true MWEs and tagging only the head nouns thereof leads to partial credits.

By restricting patterns to coordinations, Basilisk-C avoids MWE recognition problems. Coordinated noun phrases tend to be less modified, less complex and the context of an NP within the coordination makes it easier to determine its boundaries; the internal boundaries are always connectors. Table 5 shows some MWEs induced by Basilisk-G and the subparts thereof induced by Basilisk-C.

<p>high-molecular weight <u>vinylidene fluoride resin</u>  above <u>metal chelate compound</u>  unsaturated <u>fatty acid ester</u>  heat-fusible <u>polymer fine particle</u></p>
--

Table 5: Examples of MWEs in B-G; underlined tokens match MWEs induced by B-C

**Coordination abundance.** Basilisk-C works best for a text type in which large coordinations are abundant since this is the only context pattern it considers. In an analysis of the prevalence

of coordinations in different corpora, we observed that long coordinations (those with at least three conjuncts) are more prevalent in patents than in other genres (average length of 4.6 in EPO vs 3.6 in other corpora). Thus, coordinations seem to be a particularly promising resource for lexical bootstrapping in technical domains like patents. However, as exemplified by our experiments with Wikipedia, Basilisk-C shows similar performance on other domains, given that the members of the semantic class appear often in coordinations.

## 6 Related work

We have chosen a semisupervised approach to lexical bootstrapping here since it is reasonable to expect that in the type of application scenario we have in mind resources are available to create a seed set. There are also completely unsupervised approaches to lexical bootstrapping (e.g., Lin and Pantel (2002); Davidov et al. (2007); Van Durme and Paşca (2008); Dalvi et al. (2012)), but they usually cannot match the quality of approaches like ours that use human input such as a seed set.

The bootstrapping approach we have adopted here starts with a seed set and then iteratively extends the lexicon by adding the highest-confidence MWEs in each iteration. Basilisk (Thelen and Riloff, 2002) is perhaps the best known bootstrapping method of this type, but there exists a large literature on similar methods, some of which exploit lexical co-occurrence statistics (e.g., Riloff and Shepherd (1997)) and some of which use syntactic analysis (e.g., Roark and Charniak (1998); Riloff and Jones (1999); Phillips and Riloff (2002)). Our approach does not make use of syntactic analysis but relies on POS patterns.

Some recent work attempts to improve Basilisk’s accuracy. Igo and Riloff (2009) enhance precision by checking candidate terms using web queries. Qadir and Riloff (2012) combine Basilisk in an ensemble with an SVM tagger and a coreference resolution system. Our focus is learning technical terminology from very large corpora using coordinations, but any work that improves the accuracy of basic Basilisk could also be beneficial in our setting.

Gazetteers are crucial for good performance in named entity recognition (NER). Work on automatic extraction of gazetteers for NER includes (Toral and Muñoz, 2006; Kazama and Torisawa, 2007). Most of this work is complementary to

our approach because it uses knowledge bases like Wikipedia or is only applicable to traditional named entities (NEs). Traditional NEs like person are capitalized. Substances are not. Our work also differs in its focus on coordinations and technical text.

Coordinations have been frequently used in work on lexical acquisition. Caraballo (1999) builds a hierarchy of coordinated nouns and their hypernyms. Cederberg and Widdows (2003) use coordinations to estimate the semantic relatedness of nouns. Widdows and Dorow (2002) and Qiu et al. (2011) cluster nouns and evaluate the semantic homogeneity of the clusters. Etzioni et al. (2005) use Hearst patterns to bootstrap lexicons. They also consider coordinations when selecting candidates. This previous work on coordinations is unsupervised and not focused on learning a particular semantic class that is defined by a seed set.

Roark and Charniak (1998) use a variety of syntactic constructions, including coordinations, for bootstrapping. Our approach is different in that we do not require parsing and that we cover MWEs in general, not just heads or compounds with a common head. However, some of the other syntactic constructions presented by Roark and Charniak (1998) could also be amenable to reliable detection by REs. We plan to investigate this in future work. Goyal et al. (2010) create a plot unit representation creator. Therefore, they induce a lexicon of *patient polarity verbs* (i.e., verbs that impart positive or negative states on their patients) based on Basilisk, that learns from coordinated verbs. This work is focused on verbs with the same patient polarity in binary coordinations extracted from a web corpus. Our approach is based on coordinations of any size from a large patent corpus and focuses on semantic lexicon induction.

One distinguishing characteristic of our work is the patent domain. Other work on technical or scientific domains includes press releases of pharmaceutical companies (Phillips and Riloff, 2002), medline abstracts (McIntosh and Curran, 2009), message board posts from the Veterinary Information Network (Huang and Riloff, 2010) and texts from ProMed and PubMed (Igo and Riloff, 2009; Qadir and Riloff, 2012).

Patents can be argued to be particularly difficult technical text due to long sentences, legalese and complex NP syntax. To the best of our knowledge, our experiments are also the largest seman-

tic bootstrapping experiments on technical text to date. While there has been much work on experiments on large web corpora and other general text (e.g., Kozareva et al. (2008); Carlson et al. (2009); Bakalov et al. (2011)), the corpora in other lexical bootstrapping work on technical domains have been an order of magnitude smaller than ours.

We showed that using only coordinations remedies the problem of semantic drift. Other work on semantic drift includes Yangarber et al. (2002); Curran et al. (2007); McIntosh and Curran (2008); McIntosh and Curran (2009).

## 7 Conclusion

In this paper, we presented Basilisk-C. The method is inspired by original Basilisk but adapts it to large corpora of technical text by restricting it to one type of patterns: coordinations.

This restriction to coordinations, a relation that is known to impose strong semantic coherence upon its members and as such a possible remedy for semantic drift, leads to significant improvements for the task of semantic tagging, compared to an unrestricted version of Basilisk.

We further extended original Basilisk to include MWEs, as these are predominant in technical text and showed that coordination patterns yield higher precision in MWE extraction.

The proposed method avoids the need for parsing, which is cumbersome for large corpora with long sentences, typical for the technical domain. In general, we upgrade scalability because the coordination patterns represent a fraction of the patterns original Basilisk utilized.

Apart from using linguistics patterns such as coordinations, we plan to use structured data, such as table columns and rows to extract co-hyponyms in future work.

We will make our gold-standard and the induced lexicons publicly available<sup>8</sup>.

## Acknowledgments

This research was supported by the project TOPAS<sup>9</sup> (Tool platform for intelligent Patent Analysis and Summarization), its team members and the European Commission with its FP7-SME Program. The TOPAS Consortium is composed of five partners: Brüggmann Software Inc., IALE Inc., Intelisemantic Inc., Pompeu Fabra University and University of Stuttgart. We thank all colleagues for their support. This research was further supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) as part of the SFB 732.

<sup>8</sup>[www.ims.uni-stuttgart.de/data/basiliskc.resources.tgz](http://www.ims.uni-stuttgart.de/data/basiliskc.resources.tgz)

<sup>9</sup>[topasproject.eu](http://topasproject.eu)

## References

- A. Bakalov, A. Fuxman, P. P. Talukdar, and S. Chakrabarti. 2011. Scad: Collective discovery of attribute values. In *WWW 2011*.
- B. Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING 2010*.
- S. A. Carballo. 1999. Automatic acquisition of a hypernym-labeled noun hierarchy from text. In *ACL 99*.
- A. Carlson, J. Betteridge, E. R. Hruschka, and T. M. Mitchell. 2009. Coupling semi-supervised learning of categories and relations. In *NAACL-HTL 2009*.
- S. Cederberg and D. Widdows. 2003. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *CoNLL 2002*.
- M. Ciaramita and M. Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *EMNLP 2003*.
- J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20:37–46.
- J. R. Curran, T. Murphy, and B. Scholz. 2007. Minimising Semantic Drift with Mutual Exclusion Bootstrapping. In *PACLING 2007*.
- B. Dalvi, W. W. Cohen, and J. Callan. 2012. Websets: Extracting sets of entities from the web using unsupervised information extraction. In *WSDM 2012*.
- D. Davidov, A. Rappoport, and M. Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *ACL 2007*.
- O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165:91–134.
- R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. 2003. Named entity recognition through classifier combination. In *HLT-NAACL 2003*.
- A. Goyal, E. Riloff, and H. Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *EMNLP 2010*.
- R. Huang and E. Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *ACL 2010*.
- S. P. Igo and E. Riloff. 2009. Corpus-based semantic lexicon induction with web-based corroboration. In *NAACL 2009*, pages 18–26.
- J. Kazama and K. Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL 2007*.
- Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL 2008*.
- J. R. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.
- D. Lin and P. Pantel. 2002. Concept discovery from text. In *COLING 2002*.
- T. McIntosh and J. R. Curran. 2008. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *ALTA 2008*.
- T. McIntosh and J. R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *ACL-IJCNLP 2009*.
- W. Phillips and E. Riloff. 2002. Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In *EMNLP 2002*.
- A. Qadir and E. Riloff. 2012. Ensemble-based semantic lexicon induction for semantic tagging. In *\*SEM-2012*.
- L. Qiu, Y. Wu, J. Shi, Y. Shao, and Z. Long. 2011. Induction of semantic classes based on coordinate patterns. In *WI-IAT 2011*.
- L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL 2009*.
- E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI-99*.
- E. Riloff and W. Phillips. 2004. An introduction to the sundance and autoslog systems. Technical report.
- E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *EMNLP 1997*.
- B. Roark and E. Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *ACL 98*.
- H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, September.
- M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *EMNLP 2002*.
- A. Toral and R. Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *EACL 2006*.
- B. Van Durme and M. Paşca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *AAAI 2008*.
- D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *COLING 2002*.
- R. Yangarber, W. Lin, and R. Grishman. 2002. Unsupervised learning of generalized names. In *COLING 2002*.
- A. Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000*.

# Long-Distance Time-Event Relation Extraction

**Alessandro Moschitti**

QCRI, Qatar Foundation, Doha, Qatar  
DISI, University of Trento, Povo (TN), Italy  
amoschitti@qf.org.qa  
moschitti@disi.unitn.it

**Siddharth Patwardhan and Chris Welty**

IBM T. J. Watson Research Center  
Yorktown Heights NY 10598  
siddharth@us.ibm.com  
welty@us.ibm.com

## Abstract

This paper proposes state-of-the-art models for time-event relation extraction (TERE). The models are specifically designed to work effectively with relations that span multiple sentences and paragraphs, i.e., inter-sentence TERE. Our main idea is: (i) to build a computational representation of the context of the two target relation arguments, and (ii) to encode it as structural features in Support Vector Machines using tree kernels. Results on two data sets – Machine Reading and TimeBank – with 3-fold cross-validation show that the combination of traditional feature vectors and the new structural features improves on the state of the art for inter-sentence TERE by about 20%, achieving a 30.2 F1 score on inter-sentence TERE alone, and 47.2 F1 for all TERE (inter and intra sentence combined).

## 1 Introduction

Time-Event Relation Extraction (TERE) is the task of linking event mentions and relation mentions to occurrences of “time stamps” in text. We define it as follows: given a set of textual expressions denoting events and relations, and a set of time expressions in the same text document, find all instances of temporal relations between elements of the two input sets. A relation between an event and a time expression indicates that the event occurs within the temporal context specified by the time expression, for example, the following sentence:

*He succeeded James A. Taylor, who stepped down as chairman, president and chief executive in March for health reasons; the appointment took effect Nov. 13.*

conveys two different events, *succession* and *stepping down*, linked to the time stamps, *Nov. 13* and *March*, respectively.

In this paper, we focus on the task of linking time expressions to events, i.e., we carry out a classification task, where, for each possible pair of (event/relation, time) in a document, the classifier decides whether there exists a link between the two. In particular, we assume that the event mentions, relation mentions and time expressions are given to us by an external process. There is a large body of work on the above topics and they remain difficult problems, but we use human annotated mentions and expressions as input to our models since TERE itself is a relatively new problem in this context. Previous work in TempEval-2 (Verhagen et al., 2010) and our work (Hovy et al., 2012) have shown that accurate relation classifiers can be modeled with supervised approaches, provided that the expressions are limited to be in the same sentence. In contrast, there is almost no previous work on inter-sentence TERE (ISTERE), for three main reasons:

- Across a document, the number of time-event pairs to consider is large, as they are quadratic in the number of time and event expressions.
- There are almost no practically useful linguistic models that can be applied for capturing inter-sentence relations.
- Defining inter-sentence features is complex: their non-optimal definition in a task such as TERE – where there is a rather high imbalance between positive and negative examples – results in underperforming machine learning models.

In this paper, we design novel supervised models for ISTERE based on a structural representation of the pairs of sentences that contain the target rela-

tion arguments. We define methods to deal with time-event relations, where the text fragment indicating the time expression, e.g., *the appointment took effect Nov. 13* of the example above, is separated from the main event, e.g., *succession and stepping down*. In particular, our representation is constituted by a pair of shallow syntactic trees (one for each sentence containing the relation arguments), where their nodes are enriched with semantic labels, i.e., EVENT and TIME. We rely on automatic feature engineering with structural kernels (see e.g., (Moschitti, 2008; Moschitti, 2009)) to feed the learning algorithm with meaningful patterns implicitly described by such a representation. Kernels are applied to our shallow syntactic representations of text resulting in a model robust to noise and easily adaptable to new domains and tasks, such as ISTERE.

We tested our models on Machine Reading and TimeBank datasets over three different configurations: (i) relation arguments both within the same sentence, (ii) relation arguments in different sentences and (iii) relation arguments both, within and across, sentences. Our experiments demonstrate that such approach is very promising, as it improves over the state of the art for ISTERE by up to 20% in F1.

In the remainder of the paper, Sec. 2 surveys the related work, Sec. 3 presents the previous state-of-the-art models for intra-sentence TERE also using structural kernels, Sec. 4 describes our new models for intra/inter TERE, Sec. 5 lays out the experiments and, finally, Sec. 6 discusses the results deriving our conclusions.

## 2 Related Work

The extraction of relations between entities has been a long-standing topic of research, with work spanning more than a couple of decades, e.g., ACE (Doddington et al., 2004) and MUC (Grishman and Sundheim, 1996).

In particular, sentence-level Relation Extraction (RE) has been typically modeled with supervised approaches, using manually annotated data, such as ACE (Kambhatla, 2004). Most work has focused on kernel methods, i.e., string and tree kernels (Bunescu and Mooney, 2005; Culotta and Sorensen, 2004; Zhang et al., 2005; Zhang et al., 2006) or their combinations (Nguyen et al., 2009). From the kernel perspective, our approach to TERE is another variant of the general RE work using kernel: we use PTK applied to two-level

shallow syntactic trees, which extracts a sort of hierarchical subsequences. This follows up our rather long research, e.g., tree kernels for modeling the relations between syntactic constituents embedded in pairs of text (i.e., question and answer passage) for answer re-ranking (Moschitti et al., 2007; Moschitti, 2008; Moschitti, 2009; Moschitti and Quarteroni, 2008; Moschitti and Quarteroni, 2010). A more computationally expensive solution based on enumerating relational links between constituents was given in (Zanzotto and Moschitti, 2006; Zanzotto et al., 2009) for the textual entailment task. Some faster versions were provided in (Moschitti and Zanzotto, 2007; Zanzotto et al., 2010). More efficient solutions based on a shallow tree and relational tags were recently proposed in (Severyn and Moschitti, 2012; Severyn et al., 2013).

Regarding the more specific task of extraction of temporal relations, the typical approaches follow similar principles of the above RE methods. Early work was devoted to ordering events with respect to one another, e.g., (Chambers and Jurafsky, 2008), and detecting their typical durations, e.g., (Pan et al., 2006). The TempEval workshops (Verhagen et al., 2007) defined the task of (i) extracting temporal relations between events and time expressions and (ii) naming relations like BEFORE, AFTER or OVERLAP. We focus on the first part of the TempEval task, following (Filatova and Hovy, 2001; Boguraev and Ando, 2005; Hovy et al., 2012), where we used the the system and results associated with the latter paper as a baseline of this paper. (Mirroshandel et al., 2011) used syntactic tree kernels for event-time links in the same sentence. As we aim at exploring long-distance RE, we consider more robust representations than syntactic trees, i.e., shallow syntactic trees, which we have successfully used in other research, e.g., (Severyn and Moschitti, 2012).

A recent challenge, i2b2<sup>1</sup> 2012, also dealing with ISTERE was carried out in the biomedical domain. We could not directly compare with the challenge's systems as their results were not available to us during the writing of this paper. Thus, we can only report on work targeting similar tasks, e.g., (Mani et al., 2006) used time relations between events to build a classifier that marks each pair of events with a temporal relation, exploiting temporal closure properties; and (ii) (Kolomiyets

<sup>1</sup><https://www.i2b2.org/pubs/index.html>



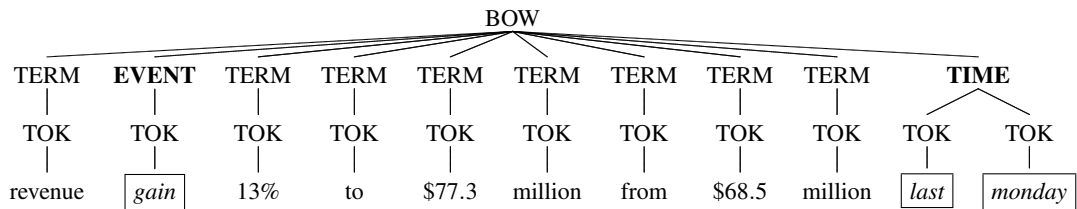


Figure 1: A kernel representations of the baseline model constituted by a bag-of-words (BOW) tree

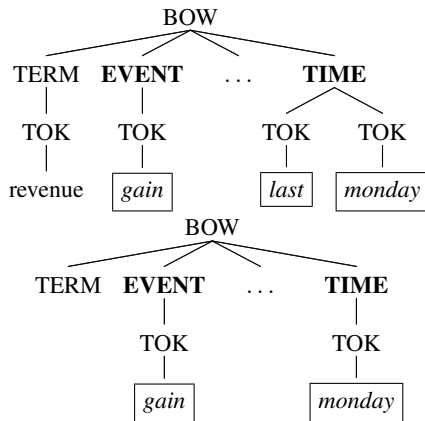


Figure 2: E.g. features from STK with BOW tree

et al., 2012) proposed to link events via partial ordering relations like BEFORE, AFTER, OVERLAP and IDENTITY.

Finally, a recent work explicitly tackling the ISTERE task is described in (Do et al., 2012). Their system was based on three classifiers: (i) a local classifier, which processes all pairs of events and time expressions in a document and decides which pairs are linked together; (ii) a classifier between pairs of events, which determines their relations: BEFORE, AFTER, OVERLAP and NO RELATION; and (iii) a joint model which, exploiting global constraints, can highly improve the overall ISTERE accuracy. We will further comment on this work in Sec. 6.

### 3 Baseline Models for TERE

The analysis of previous work has shown that there is almost no models for ISTERE. Therefore to align with prior work, we compare with previous models on *standard* TERE (extraction within a sentence). For this purpose, after formally defining the task, we describe the system we used as our baseline, which includes two types of features: (i) those manually designed also called linear features and (ii) structural features generated by tree kernels.

**Task Definition.** TERE is formally defined as follows: given the sets of expressions  $E$  denoting events or relation mentions, and  $T$  describing

time expressions in the same document: (i) build all pairs  $\langle e, t \rangle$  where  $e \in E$  and  $t \in T$ ; and (ii) classify  $\langle e, t \rangle$  to determine if a time-event relation is held, i.e., if  $e$  occurs or holds within the temporal context specified by  $t$ . In our study, we assume that: (i) a timestamp must be explicitly stated for each event/relation that we consider to be in a temporal relation; and (ii) every event/relation is associated with only one time expression whereas a temporal expression can be linked to one or more events or relations.

**Feature Vectors.** We used system and features defined in our previous work (Hovy et al., 2012), which in turn are based on the work by (Boguraev and Ando, 2005). The feature set can be divided in three different classes: (i) Features associated with events or relations. These are very similar to features typically used to represent the context of entities in traditional relation extraction tasks, which are primarily syntactic features drawn from the parser and for reporting verbs. (ii) Features specific to the temporal expressions. These are primarily designed to capture various properties of the temporal expressions. For instance, whether it is a duration, time or date, or whether its premodifiers are among those that indicate the type of expression. We include also surface features, such as numeric or non-numeric tokens in the phrase. (iii) Features describing context around both the arguments. These are primarily drawn from the work by Boguraev and Ando, and include features such as n-grams and syntactic/structural patterns. The latter also cover syntactic relations between an event and a temporal expression, ordering of the two in the sentence, etc.

**Tree Kernels.** Convolution tree kernels (TK) compute the number of common substructures between two trees without explicitly considering the whole fragment space. TKs are equivalent to a scalar product between vectors of tree fragments. Therefore using TK in SVMs is equivalent to use subtrees as features. Different TKs exist, here we consider the partial tree kernel (PTK) defined in

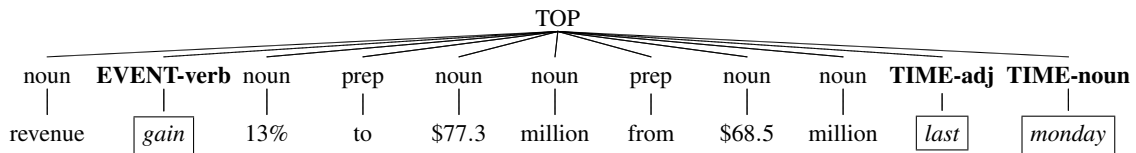


Figure 3: New sentence tree representation (STR)

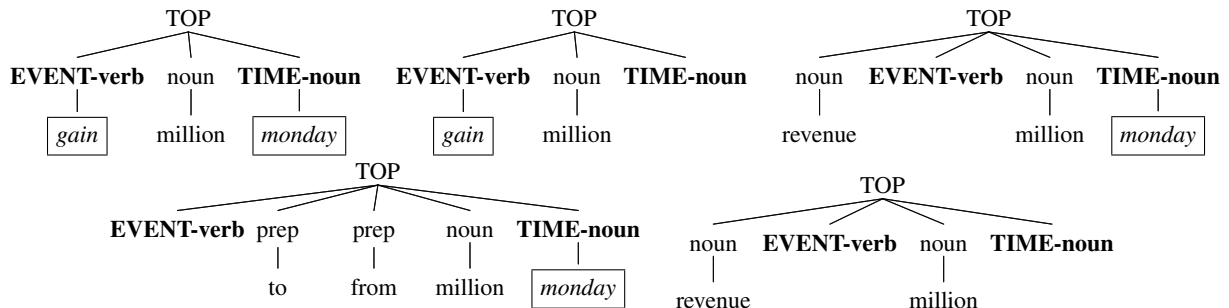


Figure 4: Some of the structural features generated by PTK when applied to the STR of Fig. 3

(Moschitti, 2006), which can count any tree fragments constituted by connected nodes.

**Computational Structures.** In (Hovy et al., 2012), we showed that combining manually engineered features with tree kernels produces a better model. We also show that exploiting syntactic information of the sentence containing the relational expressions is not trivial. Thus, we developed a bag-of-words (BOW) tree representation capturing the context of the target time/event expressions. The latter were marked-up with labels such as EVENT (or equivalently RELATION) and TIME. Figure 1 illustrates an example of the tree representations for the sentence:

*Revenue gained 13% to \$77.3 million from \$68.5 million last Monday.*

Such BOW tree is constituted by: (i) a root node; (ii) a conceptual level, which specifies the semantic type in the sentence, i.e., TERM, TIME or EVENT expressions; (iii) a token node level (TOK); and the lexical level, listing the words of the constituents. PTK applied to such trees generates features, such as [TIME [last][monday]] and [TIME [[monday]]] or more interesting features like those shown in Figure 2. For example, such features can learn the pattern: *revenue, EVENT\_gain, \*, last, TIME.*

It should be noted that such a tree represents only one relation. In case a sentence contains more than one event/relation, separate trees for each must be generated (each will be a separate training/test instance). Such trees differ in the position of the EVENT/RELATION nodes (at level

1 of the tree).

Finally, in (Hovy et al., 2012) we showed that this model significantly improves over manually engineered features. However, to exploit syntactic information, we defined another separate tree with POS-tag nodes in place of words causing the features coming from different trees to be disjoint. This model cannot be applied for ISTERE as the large number of identical nodes, TERM and TOK would cause the PTK computational complexity to degenerate to  $O(n^2)$  (see (Moschitti, 2009)).

#### 4 Models for Inter-Sentence TERE

We describe here our new representation, which is an improvement over our previous models on intra-sentence TERE and, more importantly, can be used for ISTERE.

**Intra-Sentence Representation.** We improve on the previous work by reformulating the BOW tree as follows:

1. We remove the TERM and TOK levels and we propose only two levels — the POS-tag and the word sequences.
2. The annotation of the target time or event expression is directly performed on the POS-tag node.

For example, Fig. 3 shows the transformation of the trees in Fig. 1 to the new representation. The event *gain* and the time expression *last monday* are marked at POS-tag level<sup>2</sup>. This also compacts the segmentation of time expressions. As a result, the application of PTK to the new sentence tree

<sup>2</sup>The POS tagset is the one used in the IBM Watson system (Ferrucci et al., 2010).

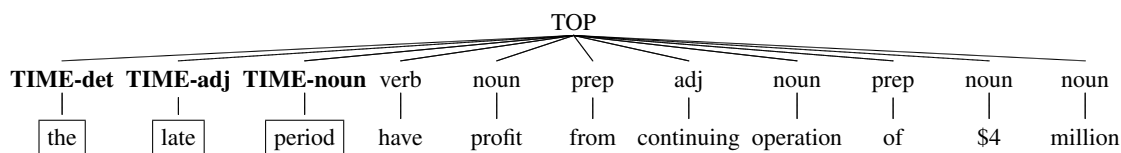


Figure 5: A second STR containing a time expression. Together with the one of Fig. 3 containing the event, it forms the representation pair, input to the kernel used for ISTERE.

representation (STR) generates very powerful and compact features, e.g., those described in Fig. 4. The last fragment of the figure suggests that if, in a sentence, there is the noun *revenue* followed by (i) any word tagged as EVENT, (ii) the noun *million* and (iii) any TIME expression, the EVENT is probably associated with such TIME expression. Note that this is not a rule but just a feature receiving a weight from SVMs based on training data.

Finally, it should be noted that the average running time of PTK will be  $O(n)$ , because the new tree does not contain many repeated node labels. This is far more efficient than the PTK applied to BOW trees (see (Moschitti, 2009)).

**Inter-Sentence Representation.** STR still cannot encode relations whose arguments are located in different sentences. Our approach for this problem is to design two STRs for the sentences containing the two potential arguments of a relation. For example, let us suppose that a time expression of the *gain*-EVENT in Fig. 3, e.g., *late period*, is expressed in the following sentence: *The late period has profit from continuing operations of \$4 million*.

We produce another STR associated with it, as shown in Fig. 5. This way, we model  $\langle e, t \rangle$  with a pair of trees  $\langle E, T \rangle$ , where  $e \in E$  and  $t \in T$  (see Sec. 3). Accordingly, we define the kernel  $K(p_1, p_2)$  over two pairs  $p_1 = \langle E_1, T_1 \rangle$  and  $p_2 = \langle E_2, T_2 \rangle$  as:  $TK(p_1, p_2) = \text{PTK}(E_1, E_2) + \text{PTK}(T_1, T_2)$ .

It should be noted that: (i) the additive combination of kernels is still a valid kernel and it corresponds to the merged fragment space of E and T trees; (ii) the kernel product can also be applied but it has shown poor results in previous work (Moschitti, 2004); and (iii) PTK allows for modeling structural features in two sentences located in different part of the document. Thus, the features will be pairs of tree fragments from E and T. It is worth noting that the pairs of BOW and POS trees used in (Hovy et al., 2012) cause PTK to be too slow for this setting (although they may achieve comparable accuracy).

Additionally, the PTK can be combined with a linear kernel of manually engineered features using an additive operation. If  $\vec{x}_i$  is the vector representation of the manually engineered features of  $p_i$ , the kernel combination of PTK and the linear kernel is  $\text{PTK}(p_i, p_j) + \vec{x}_i \cdot \vec{x}_j$ . The linear features extracted for the EVENT expression in one sentence and the TIME expression in the other sentence can reconstruct their shared context thanks to the pairs of tree fragments generated by PTK. The next section will empirically verify this hypothesis.

## 5 Experiments

In this section, (i) we compare our model against the state of the art for intra-sentence TERE; and (ii) we test its validity for ISTERE.

### 5.1 Setup

We used two corpora: Machine Reading Program (MRP) corpus to compare with our previous system (Hovy et al., 2012) (our baseline) and Time-Bank data (Pustejovsky et al., 2003), which in contrast to MRP data, enables us to train and test our system with inter-sentence gold standard (GS) annotation. During testing, we used GS annotations for the timestamps and events, i.e., we only classify which events and time expressions are linked together.

**MRP data.** Following our work in (Hovy et al., 2012), we used the data made available in MRP related to linking timestamps and events in the intelligence community (IC) domain (Strassel et al., 2010). It is based on news reports about terrorism taken from the Gigaword corpus. It includes 169 documents containing 2,046 pairs of event and temporal expressions (505 positive, 1,541 negative instances) within the same sentence. We increased the original number of event instances by means of gold event-coreference annotations, i.e., two events that co-refer will share their annotated time expressions; thus we can merge them and increase the size of our gold annotation. As before, 41% of all correct event-time pairs are not in the same sentence (for relations this ratio is more than 80% of the correct fluent-time links).

	Training Set			Validation Set			Test Set		
	Pos.	Neg.	Total	Pos.	Neg.	Total	Pos.	Neg.	Total
DIST=0	1,125	2,754	3,879	155	405	560	162	463	625
DIST>0	900	65,221	66,121	129	9,311	9,440	182	16,600	16,782
DIST≥0	2,025	67,975	70,000	284	9,716	10,000	344	17,063	17,407

Table 1: Distribution of data in the three TimeBank subsets.

While it allows us to compare with previous work, the MRP data is not very well suited for training and testing new systems modeling ISTERE. Indeed, almost all inter-sentence pairs of time/event contain members that (i) are coreference of either the time or the event and (ii) are typically located in the same sentence. This means that, given an accurate system for intra-sentence TERE and a good coreference resolution system, ISTERE described in the MRP corpus can be easily solved. The combined system is still very complex and interesting, but it inevitably falls in the class of coreference resolution problems. Here we aim at studying linguistic phenomena directly connected to inter-sentence relations, which go beyond coreference resolution. For this reason, we also ran experiments on a second corpus described below, which is more suitable to our study.

**TimeBank corpus.** Distributed by the Linguistic Data Consortium<sup>3</sup>, it consists of 183 documents – news articles from several news sources that have been annotated with event and time expressions compliant with the TimeML specification<sup>4</sup>. We divided the corpus in three subsets containing relations whose arguments are located in: (i) the same sentence (DIST=0), (ii) more than one sentence (DIST>0) and (iii) both previous cases (DIST≥0). The distribution of positive and negative examples in the training, validation and test sets are reported in Table 1. It is interesting to note that the distribution of positive examples in DIST=0, i.e., the intra-sentence relations, is 30% of all possible pairs. In contrast, such distribution in DIST>0, i.e., inter-sentence relations, drastically reduces to 1.4% of the pairs occurring in a document. This imbalance immediately gives the feeling of the complexity of the ISTERE task.

**Learning Model.** we used SVM-Light-TK (Moschitti, 2006; Joachims, 1999), which enables the use of the Partial Tree Kernel (PTK) (Moschitti, 2006). We used the default kernel hyperparameters and the margin/error trade-off param-

<sup>3</sup>LDC2006T08 at <http://www.timeml.org/site/timebank/documentation-1.2.html>

<sup>4</sup>The inter-annotator agreement numbers are specified in the referring website

ter to favor replicability of our results, and study instead the cost-factor parameter as it tunes the balance between Precision and Recall, which is very critical for our task (highly skewed datasets). **Measures.** We estimated Precision, Recall and F1 with 10-fold cross-validation for MRP experiments for comparing with (Hovy et al., 2012). For TimeBank, we drew F1 plots using the random test set described in Table 1. To estimate the final F1 of our models, we used 3-fold cross-validation<sup>5</sup> applied to the merged train and test set in the table. In this case the cost-factors were estimated from the validation set (see the table above), which is not part of the merged data used for the 3-fold cross-validation. It should be noted that to create the folds and the other subsets, we took care to not mix RE pairs between the folds or between training, validation and test set.

## 5.2 Intra-Sentence TERE: MRP Results

We trained SVMs using PTK applied to STR of single sentences and also combined with linear features. We tested a few parameter values of the Precision/Recall trade-off (cost-factor) on a validation set then, following (Hovy et al., 2012), we ran 10-fold cross-validation. The average F1 was 76.84, which is directly comparable with the outcome we reported in (Hovy et al., 2012), i.e., an F1 of 76.5 (when linear features are used in combination with tree kernels). It should be noted that: (i) in (Hovy et al., 2012) we showed that our system achieved the same accuracy than the best system of TempEval-2; and (ii) our STR provides the same results of the combination of the two structures we used in the model above.

Additionally, we tested PTK alone and attained an F1 of 74.45. This basically suggests that if we only use tree kernels, we can trade-off several months of work for manual feature engineering for a little bit less accurate system. Those unfamiliar with structural kernels may think that the time spent for engineering tree representations is comparable to the one spent for engineering features.

<sup>5</sup>It is more suitable than a 10-fold setting for deriving the final accuracy, given the very low number of positive examples in DIST>0.

However, the abstraction provided by the tree kernels suggests that the effort required in engineering trees is orders of magnitudes lower. The baseline system using the manually engineered features was designed at IBM and required several months of manual effort to engineer, code and tune features. Our expert on kernel methods (who is not an expert on TERE) modeled STR in 20 minutes and the implementation only concerned the construction of strings representing trees like those in fig. 3 and 5. While this is anecdotal evidence, it is a good indicator of the power of tree kernels.

Furthermore, our experiments show that the combination of tree kernels and feature vectors is much more adaptable to variations of the TERE task and data. This can be observed, for instance, when considering RE from TimeBank.

### 5.3 Cost-factor role in ISTERE: TimeBank

We performed the first experiment using DIST=0 data and three different models, Linear (i.e., only using the manual features), PTK (which in this case only uses one tree) and their additive combination, i.e., Linear + PTK. In these experiments, the only critical parameter is the one tuning the Precision/Recall trade off (cost-factor parameter) as the high data imbalance between negative and positive examples can result in imbalanced Precision and Recall. Thus, we plotted the F1 of the above models (derived on the test set) according to a reasonable set of values of such parameter. The result is shown in Fig. 6. We note that (i) PTK produces a better F1 than linear features for any parameter value; (ii) the combination Linear + PTK highly improves on both achieving an interesting F1 of 64.35; (iii) in comparison with MRP, where the best model achieves an F1 of 76.84, the TimeBank task appears to be more difficult.

We ran an experiment for DIST>0, which considers only inter-sentence relations. Fig. 7 shows a similar curve as before, except that manual features have a higher accuracy than PTK. The F1, however, is rather low, indicating the complexity of the task and the inadequacy of manual features.

As predicted in Sec. 4, the combination of inter-sentence structural and manual features highly improves on the system F1 achieving a state-of-the-art value of 38.82 for ISTERE. Although, the result does not still guarantee a successful use of the proposed model for real-world applications, it clearly shows a promising research direction.

Finally, we tested DIST $\geq$ 0, i.e., the complete

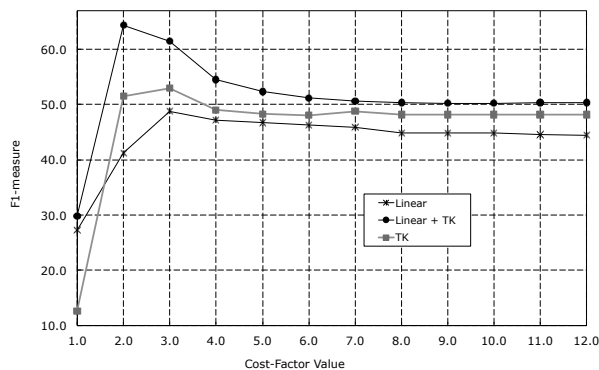


Figure 6: TERE cost-factor impact on DIST=0

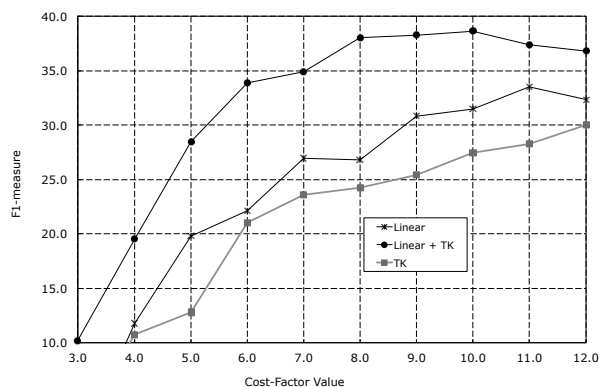


Figure 7: TERE cost-factor impact on DIST>0

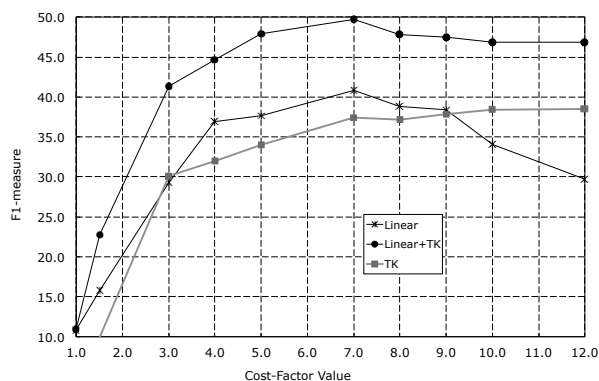


Figure 8: TERE cost-factor impact on DIST $\geq$  0

TERE task on entire documents. Fig. 8 shows comparable performance between PTK and Linear but again when combined together the improvement is rather high, i.e., up to 10 absolute percent points (25% of relative improvement on manual features). It should be noted that the above results do not indicate the final system accuracy: for this purpose, the next section shows the 3-fold cross-validation results using cost-fact values that are (i) derived from the validation set (not included in the cross-validation data) and (ii) slightly different from those optimizing the plot in the figures.

	DIST=0			DIST>0			DIST≥0		
	Linear	PTK	Lin.+PTK	Linear	PTK	Lin.+PTK	Linear	PTK	Lin.+PTK
Prec.	36.7±1.3	57.2±0.7	60.9±3.0	18.5±2.5	29.1±5.5	29.2±4.4	33.8±1.1	34.2±2.5	39.8±2.3
Rec.	89.4±3.2	42.4±1.9	64.0±2.9	55.5±5.0	20.6±8.2	32.3±7.9	57.7±1.1	46.7±1.5	58.4±4.0
F1	52.0±0.9	48.7±1.0	62.3±1.7	27.7±3.4	23.1±5.4	30.2±4.5	42.6±1.0	39.4±1.2	47.2±1.8

Table 2: 3-fold cross-validation results for DIST=0, DIST>0 and DIST≥0 tasks.

#### 5.4 Cross-Validation Results

The previous section demonstrates the superiority of the combined Linear + PTK model over manual features for any value of the cost-factor parameter. To assess the significance of this, we carried out 3-cross-fold cross validation. Table 2 shows the average Precision, Recall and F1 over the 3-folds along with the associated standard deviation (preceded by  $\pm$ ). We note that (i) the relative improvement over the Linear model derived on DIST=0, about 20%, confirms the results showed by the plots; and (ii) the relative improvement derived on DIST>0 and DIST≥0 is lower, although still remarkable, i.e., up to 9% and 12%, respectively. This is probably due to the fact that 2 folds constitute a training set of 58K instances (for DIST≥0), whereas in the plot experiments the training data contained 70K examples. Evidently the more complex patterns needed for long-distance TERE require more training data to express their entire potential. Finally, feature vectors perform better than structural kernels alone but this does not contradict the fact that kernels save potentially large engineering work since: (i) even if the features had been engineered for MRP and used for TimeBank, the effort would have been done in any case whereas kernels almost completely avoid it; and (ii) the combination largely improve on the feature vectors: this may avoid the need of additional work for feature refining.

## 6 Discussion and Conclusions

Previous work has proposed intra-sentence TERE models based on manually designed features and tree kernels. In this paper, we propose new models for inter-and intra-sentence TERE. We provided a flexible kernel, which improves efficiency and capacity of generating meaningful features. It can be applied to the pairs of all document sentences for modeling ISTERE. This enables the use of all possible pairs of tree fragments from the time and event sentences as features, which improve on the features manually designed in (Hovy et al., 2012). For example, the latter cannot capture the relation with the “document date”, when the time expressions occur in the titles. This kind of features can

be added but a study of the problem and engineering effort are required. In contrast, our models can automatically generate such features.

Our experiments on MRP and TimeBank show that our approach provides high accuracy, up to 20% of relative improvement over state of the art. The reason for such impressive results is the adaptability and automatic feature engineering properties of tree kernels. Indeed, new data and settings pose new challenges to the RE systems, which require effort in engineering both features and methods. Our approach alleviates such effort as we can use a more general-purpose technology.

In this work, our model has been applied to establish the link between time expressions and events. However, in general, our model could be applied to the complete TERE task, thus also determining the relation types. Interestingly, the model proposed in (Do et al., 2012) is based on the pairwise classifiers we study in this paper. Although, the authors used a different dataset<sup>6</sup>, which makes an exact comparison with their systems difficult, we note that their local pair classifiers achieved an F1 of 42.13 (no global model, so the same setting as ours) and an F1 of 46.01 using their global model based on ILP. Our local pairwise classifier attained an F1 of 47.2, which can be used as input to the global model to further boost the overall system accuracy.

Finally, the pairwise approach may be considered computationally expensive. However, with modern technology,  $O(n^2)$  complexity (where  $n$  is the number of sentences in a document) is feasible. PTK is efficient and can be made faster with recent reverse kernel engineering work (Pighin and Moschitti, 2010; Pighin and Moschitti, 2009).

In summary, the main message of this paper is that ISTERE is complex, requiring a significant engineering effort. We have shown that tree kernels are adaptable, requiring less effort and improving on the state of the art in the full TERE task – relations spanning more than one sentence.

<sup>6</sup>They annotated a portion of the ACE corpus with (i) event mention and time interval association, and (ii) the temporal relations between event mentions.

## Acknowledgements

This research is partially supported by the EU's 7<sup>th</sup> Framework Program (FP7/2007-2013) (#288024 LIMOSINE project) and an Open Collaborative Research (OCR) award from IBM Research.

## References

- Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of IJCAI*.
- R. Bunescu and R. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT-EMNLP*.
- N. Chambers and D. Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*.
- A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *ACL*.
- Q. Do, W. Lu, and D. Roth. 2012. Joint inference for event timeline construction. In *the joint EMNLP and CoNLL conference*.
- G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. The automatic content extraction program – tasks, data and evaluation. In *LREC*.
- D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefel, and C. Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3).
- E. Filatova and E. Hovy. 2001. Assigning time-stamps to event-clauses. In *the workshop on Temporal and spatial information processing*.
- R. Grishman and B. Sundheim. 1996. Message Understanding Conference - 6: A Brief History. In *Coling*.
- D. Hovy, J. Fan, A. Gliozzo, S. Patwardhan, and C. Welty. 2012. When did that happen? – linking events and relations to timestamps. In *EACL*.
- T. Joachims. 1999. Making large-scale SVM learning practical. *Advances in Kernel Methods – Support Vector Learning*, 13.
- N. Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *ACL*.
- O. Kolomiyets, S. Bethard, and M.-F. Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *ACL*.
- I. Mani, M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky. 2006. Machine learning of temporal relations. In *ACL*.
- S. A. Mirroshandel, M. Khayyamian, and G. Ghassem-Sani. 2011. Syntactic tree kernels for event-time temporal relation learning. *HLT*.
- A. Moschitti and S. Quarteroni. 2008. Kernels on Linguistic Structures for Answer Extraction. In *ACL*.
- A. Moschitti and S. Quarteroni. 2010. Linguistic Kernels for Answer Re-ranking in Question Answering Systems. *Information Processing & Management*.
- A. Moschitti and F. M. Zanzotto. 2007. Fast and effective kernels for relational learning from texts. In *ICML*.
- A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL*.
- A. Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *ACL*.
- A. Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*.
- A. Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *CIKM*.
- A. Moschitti. 2009. Syntactic and Semantic Kernels for Short Text Pair Categorization. In *EACL*.
- T.-V. T. Nguyen, A. Moschitti, and G. Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *EMNLP*.
- F. Pan, R. Mulkar, and J. R. Hobbs. 2006. Learning event durations from event descriptions. In *Coling-ACL*.
- D. Pighin and A. Moschitti. 2009. Reverse engineering of tree kernel feature spaces. In *EMNLP*.
- D. Pighin and A. Moschitti. 2010. On reverse feature engineering of syntactic tree kernels. In *CoNLL*.
- J. Pustejovsky, P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK Corpus.
- A. Severyn and A. Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *SIGIR*.
- A. Severyn, M. Nicosia, and A. Moschitti. 2013. Learning adaptable patterns for passage reranking. In *CoNLL*.
- S. Strassel, D. Adams, H. Goldberg, J. Herr, R. Keesing, D. Oblinger, H. Simpson, R. Schrag, and J. Wright. 2010. The DARPA MRP. In *LREC*.
- M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *the Workshop on Sem. Ev.*
- M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *the Workshop on Sem. Evaluation*.
- F. M. Zanzotto and A. Moschitti. 2006. Automatic Learning of Textual Entailments with Cross-Pair Similarities. In *COLING-ACL*.
- F. M. Zanzotto, M. Pennacchiotti, and A. Moschitti. 2009. A Machine Learning Approach to Recognizing Textual Entailment. *JNLE*.
- F. M. Zanzotto, L. Dell'Arciprete, and A. Moschitti. 2010. Efficient graph kernels for textual entailment recognition. *Fundamenta Informaticae*, 2010.
- M. Zhang, J. Su, D. Wang, G. Zhou, and C. L. Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proc. of IJCNLP*.
- M. Zhang, J. Zhang, J. Su, and G. Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *COLING-ACL*.

# Unsupervised Extraction of Attributes and Their Values from Product Description

Keiji Shinzato

Satoshi Sekine

Rakuten Institute of Technology

{keiji.shinzato, satoshi.b.sekine}@mail.rakuten.com

## Abstract

This paper describes an unsupervised method for extracting product attributes and their values from an e-commerce product page. Previously, distant supervision has been applied for this task, but it is not applicable in domains where no reliable knowledge base (KB) is available. Instead, the proposed method automatically creates a KB from tables and itemizations embedded in the product's pages. This KB is applied to annotate the pages automatically and the annotated corpus is used to train a model for the extraction. Because of the incompleteness of the KB, the annotated corpus is not as accurate as a manually annotated one. Our method tries to filter out sentences that are likely to include problematic annotations based on statistical measures and morpheme patterns induced from the entries in the KB. The experimental results show that the performance of our method achieves an average F score of approximately 58.2 points and that filters can improve the performance.

## 1 Introduction

E-commerce enables consumers to purchase a large number of products from a variety of categories such as books, electronics, clothing and foods. In recent years, this market has grown rapidly around the world. Online shopping is regarded as a very important part of our daily lives.

Structured product data are most important for online shopping malls. In particular, product attributes and their values (PAVs) are crucial for many applications such as faceted navigation and recommendation. However, since structured information is not always provided by the merchants, it is important to build technologies to create this

structured information (such as PAVs) from unstructured data (such as a product description). Because of the wide variety of product types, such technology should not rely on a manually annotated corpus. One of the promising methods for information extraction (IE) without a manually annotated corpus is *distant supervision* (Mintz et al., 2009), which leverages an existing knowledge base (KB) such as Wikipedia or Freebase to annotate texts using the KB instances. These popular KBs, however, are not very helpful for distant supervision in an e-commerce domain for the following reasons. (1) An infobox in a Wikipedia article is not always tailored towards e-commerce. For instance, as of May 2013, the infobox attributes of wine in English Wikipedia included energy, carbohydrates, fat, protein and alcohol<sup>1</sup>. These are not particularly useful for users seeking their favorite wines through online shopping. Instead, the grape variety, production area, and vintage of the wine would be of greater interest. (2) On the other hand, Freebase contains PAVs for limited types of products such as digital cameras<sup>2</sup>. However, since Freebase is currently only available in English, we cannot use Freebase in a distant supervision method for other languages. Moreover, the number of categories whose PAVs are available in Freebase is limited even in English.

In this paper, we propose a technique to extract PAVs using an automatically induced KB. For the induction, the method uses structured data such as tables and itemizations embedded in the unstructured data. An annotated corpus is then automatically constructed from the KB and unstructured data, i.e. product pages. Since these texts are in HTML format, we can extract the attribute candidates and their values using pattern matching with tags and symbols. We can expect the KB to

<sup>1</sup><http://en.wikipedia.org/wiki/Wine>

<sup>2</sup>[http://www.freebase.com/view/digicams/digital\\_camera](http://www.freebase.com/view/digicams/digital_camera)



have a certain level of accuracy because the tables and itemizations are created by merchants who are product experts. However, there may be a need to group synonymous attribute names. We propose a method for grouping synonymous attribute names by observing the commonality of the attribute values and their co-occurrence statistics. The model for extracting attribute values is then trained using the annotated corpus to find PAVs of the products. Because the KB is incomplete, the annotated corpus may contain annotation mistakes; *false-positive* and *false-negative*. Thus, our method tries to filter out sentences that are likely to include those problematic annotations based on statistical measures and morpheme patterns induced from the entries in the KB.

The contributions of our work are as follows:

1. Unsupervised and scalable methods for inducing a KB consisting of PAVs, and for discovering attribute synonyms.
2. Unsupervised method for improving the quality of an automatically annotated corpus by discarding false-positive and false-negative annotations. These problematic annotations are always included in automatically annotated corpora although such corpora can be constructed without requiring expensive human effort.
3. Comprehensive evaluation of each component: Automatically induced KBs, annotation methodology, and the performance of IE models based on the automatically constructed corpora. In particular, the annotation methodology and the IE models are evaluated by using a dataset comprising 1,776 manually annotated product pages gathered from eight product categories.

To summarize, as far as we know, this is a first work to extract product attributes and their values relying solely on product data, and completing all the steps of this task, including KB induction, in a purely unsupervised manner.

## 2 Related Work

### 2.1 Product Information Extraction

Bing et al. (2012) proposed an unsupervised methodology for extracting product attribute-values from product pages. Their method first generates word classes from product review data re-

lated to a category using Latent Dirichlet Allocation (LDA) (Blei et al., 2003). The method then automatically constructs training data by matching words in classes and those in product pages for the category. After that, extraction models are built using the training data. Their method does not deal with attribute synonymy and problematic annotations in their training data. They evaluated the result of their extraction model only, while this paper reports on evaluation results of not only extraction models but also induced KBs and annotation methodology. In addition, their method employs LDA for generating word classes. This may involve the issue of *scalability* when running the method on large size real-world data. On the other hand, we employ a simple rule-based approach to induce the KBs. We can then straightforwardly apply the method on the large-scale data.

Mauge et al. (2012) also proposed methods to extract product attribute-values using hand-crafted patterns, and to group synonymous attributes in a supervised manner. They, however, only evaluated a part of the extracted attribute names, and aggregated synonymous attribute names. They did not evaluate the extracted attribute values.

Our work is also similar to Ghani et al. (2006), who construct an annotated corpus using a *manually tailored* KB and then train models using the corpus to extract attribute values. Probst et al. (2007) and Putthividhya and Hu (2011) also proposed a similar approach with the work of Ghani. The main difference between these works and ours is that our method does not require manually tailored KBs. Instead, our method automatically induces a KB of PAVs from structured data embedded in product pages.

In addition to the above, many wrapper based approaches have been proposed (Wong et al., 2008; Dalvi et al., 2009; Gulhane et al., 2010; Ferrez et al., 2013). The goal of these approaches is to extract information from documents semi-structured by any mark up language such as HTML. On the other hand, our method aims at extracting (product) information from full texts although the method leverages semi-structured documents to induce KBs.

### 2.2 Knowledge Base Induction

There are many works for automatically inducing KBs using syntactic parsing results (Rooth et al., 1999; Pantel and Lin, 2002; Torisawa, 2001; Kazama and Torisawa, 2008), and semi-

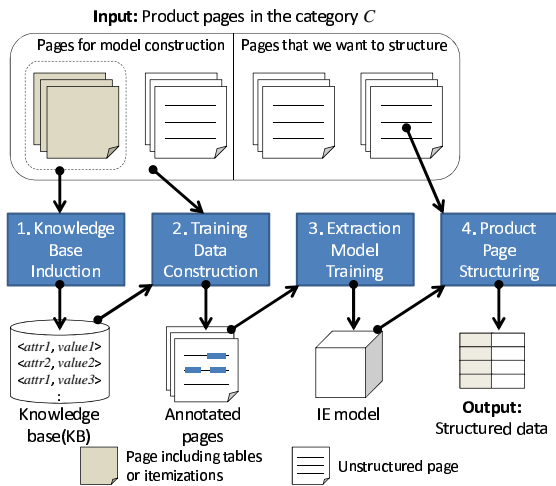


Figure 1: Overview of our approach.

structured data (Shinzato and Torisawa, 2004; Wong et al., 2008; Yoshinaga and Torisawa, 2006; Mauge et al., 2012). As clues for the induction of attribute-value KBs, previous works employ lexico-syntactic patterns such as *<attribute>:<value>* and *<value> of <attribute>*, and layout patterns in semi-structured data such as tables and itemizations marked up by HTML tags.

We employ similar clues as Yoshinaga and Torisawa (2006) for KB induction. In addition to the induction, our method tries to find synonymous attributes in the induced KB. This is the difference between Yoshinaga’s work and ours. The method of Mauge et al. (2012) also finds attribute synonyms, but it requires supervision to find these.

### 3 Data

We used Rakuten’s product pages comprising over 100 million products in 40,000 categories<sup>3</sup>. Each product is assigned to one category by merchants offering it on the Rakuten site. In contrast to Amazon, Rakuten’s product pages are not well-structured, because the product pages are designed by the merchants. Although some pages include tables for describing product information, majority of pages describe product information in full texts. That’s why we need to extract product information from the texts.

### 4 Approach

An overview of our approach is shown in Figure 1. The input for the approach is a set of product pages belonging to a single category like wine or shampoo. From the given pages, a value extraction model for the category is generated, and then the

<sup>3</sup><http://www.rakuten.co.jp/>

保存方法 (method of preservation), その他 (etc), 商品説明 (explanation), 広告文責 (responsibility for advertisement), 特徴 (characteristics), 仕様 (specs.)

Figure 2: List of stop words for attributes.

**P1:** <T (H|D) > [ATTR] </T (H|D) ><TD> [ANY] </TD>  
**P2:** [P] [ATTR] [S] [ANY] [P]  
**P3:** [P] [ATTR] [ANY] [P]  
**P4:** [ATTR] [S] [ANY] [ATTR] [S]

Figure 3: Patterns for extracting attribute values. The [ATTR] tag denotes a collected attribute, the [ANY] tag denotes a string, the [P] tag denotes the prefix and *open*-braces, and the [S] tag denotes the suffix and *close*-braces. The prefix, suffix and brace are defined in (Yoshinaga and Torisawa, 2006). Attributes of HTML tags are removed before running **P1**.

model is used to structure unstructured pages in the category.

A detailed explanation of steps 1, 2 and 3 in Figure 1 is given in the remaining sections. Explanation of step 4 is omitted because the extraction model is simply applied to a given page.

#### 4.1 Knowledge Base Induction

##### 4.1.1 Extraction of Attribute-Values

A KB is induced from the tables and itemizations embedded in the given product pages. We first collected product attributes based on the assumption that *expressions that are often located in table headers can be considered as attributes*. The regular expression `<TH.*?>.+?</TH>` is run on the given pages, and expressions enclosed by the tags are collected as attributes. (The `<TH>` tag is used for a table header.) The collected candidate set includes expressions that can not be regarded as attributes. We excluded these using a small set of stop words given in Figure 2.

To extract values corresponding to the collected attributes, the regular expressions listed in Figure 3 are run on the given product pages. An expression that matches the position of [ANY] is extracted as a value of the attribute corresponding to [ATTR]. The first appearance of [ATTR] is selected as the attribute in **P4**. The extracted value and its attribute are stored in the KB along with the number of merchants that use these in the table and itemization data. Henceforth, we refer to this number as the *merchant frequency* (MF).

##### 4.1.2 Attribute Synonym Discovery

The KBs constructed in the previous section still contains numerous synonyms; that is, attributes

with the same meaning, but different spellings, are included. This is because merchants do not have a standard method for describing products on their own pages. For example, “Bordeaux” and “Tuscany” can be regarded as production areas of wine. However, some merchants refer to “Bordeaux” as the *production area*, while others consider “Tuscany” to be the *region*. If we use KBs that include the incoherence as an annotation resource, corpora containing annotation incoherence are constructed. To avoid this problem, we set out to identify synonymous attributes in the KBs.

We attempt to discover attribute synonyms according to the assumption that *attributes can be seen as synonyms of one another if they have never been included in the same structured data, and they share an identical popular value*. We regard the MF of a value as a measure of *popularity*. Based on this assumption, for all combinations of attribute pairs with an identical attribute value whose MF exceeds  $N$ , we verify whether they appear simultaneously in structured data (i.e., table and itemization data). Attribute pairs satisfying the condition are regarded as synonyms. The value of  $N$  is defined by the equation  $N = \max(2, M_S/100)$  where  $M_S$  is the number of merchants providing structured data in the category. As a result, we obtain a vector of attributes that can be regarded as synonyms, such as (*region, country*), and (*grape, grape variety*) for the wine category. We refer to this set of synonym vectors as  $S_{attr}$ .

Next, synonym vectors with high similarity are iteratively aggregated. For example, the vector (*region, country, location*) is generated from the vectors (*region, country*) and (*location, country*). The similarity between vectors  $v_a$  and  $v_b$  is computed by the cosine measure:  $sim(v_a, v_b) = v_a \cdot v_b / |v_a| |v_b|$ . We iteratively aggregate the two vectors with the highest similarity in set  $S_{attr}$ . The aggregation process continues until the highest similarity is below the threshold  $th_{sim}$ , which we set to 0.5. The threshold value was determined empirically.

Table 1 shows an example of the KB for the wine category. We can see that the “variety” attribute includes three Japanese variants.

## 4.2 Training Data Construction

An annotated corpus for training a model for attribute value extraction is constructed from the given product pages and the KB built from those

Table 1: Example of the KB for the wine category. The top three attributes are listed with their top four values. (*number*] denotes the MF of a value.)

Attribute	容量 (volume)	品種 (variety)	タイプ (type)
Attribute synonyms	内容量 (content)	ぶどう品種 (grape variety) ブドウ品種 (grape variety) 使用品種 (usage variety)	—
Attribute values	750ML[147] 720ML[64] 375ML[49] 500ML[41]	シャルドネ [59] (Chardonnay) メルロー [36] (Merlot) シラー [29] (Syrah) リースリング [29] (Riesling)	辛口 [34] (dry) 赤 [24] (red) 白 [23] (white) フルボディ [23] (full body)
# values	159	1,578	153

pages. Values in the KB are simply annotated for the pages. Then, sentences possibly including incorrect annotations or sentences where annotation are missing are automatically filtered out from the annotated pages to improve annotation quality.

### 4.2.1 Attribute Value Annotation

All given product pages are split into sentences following block-type HTML tags, punctuation, and brackets. Each sentence is then tokenized by a morphological analyzer<sup>4</sup>. The longest attribute value matching a sub-sequence of the token sequence is annotated. We employed the Start/End tag model (Sekine et al., 1998) as chunk tags for the matched sequence. If the matched value corresponds to more than one attribute, the entry with the largest MF is selected for annotation. Note that if other attribute values are contained in the matched sub-sequence, they are not annotated.

### 4.2.2 Incorrect Annotation Filtering

Some attribute values with *low* MFs are likely to be *incorrect*. The quality of the corpus, and the performance of extraction models based on the corpus deteriorate if such values are frequently annotated. We detect incorrect value annotation in the corpus according to the assumption that *attribute values with low MFs in structured data (i.e., tables and itemizations) and high MFs in unstructured data (i.e., product descriptions) are likely to be incorrect*. Thus, we designed the following score:

$$Score(v) = \frac{MF_D(v)/N_M}{MF_S(v)/M_S}$$

<sup>4</sup>We used the Japanese morphological analyzer MeCab. (<http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>)

<b>T:</b>	Chateau	Talbot	is	a	famous	winery	in	France	.
<b>A:</b>	O	O	O	O	O	O	O	S-PA	O
<b>G:</b>	B-PR	E-PR	O	O	O	O	O	S-PA	O

Figure 4: Example of a sentence with missing annotations. The row **T** shows the tokens of the sentence, the row **A** shows automatic annotations, and the row **G** shows golden annotations. The ‘PR’ and ‘PA’ are abbreviations of *Producer* and *Production Area*, respectively. Label ‘O’ means that a token is not annotated with any label.

[NUMBER] [ML (milliliter)]
[シヤトー (Chateau)] [ANY]
[LOCATION] [ANY] [LOCATION] [市 (city)]

Figure 5: Examples of morpheme patterns induced from the KB for the wine category. The token [ANY] matches a sequence consisting of 1 - 3 arbitrary tokens. Tokens [LOCATION] and [NUMBER] are matched tokens whose part-of-speech is location and number, respectively.

where  $MF_D(v)$  and  $MF_S(v)$  are the MFs of value  $v$  in the product descriptions and structured data, respectively, and  $N_M$  is the number of merchants offering products in the category, and  $M_S$  is the number of merchants providing structured data in the category. The scoring function is designed so that values with a low MF for structured data and a high MF for item descriptions obtain high scores. We regard attribute values with scores greater than 30 as incorrect, and remove sentences that include annotation based on incorrect values from corpora. The threshold value was decided empirically.

#### 4.2.3 Missing Annotation Filtering

Because of the small coverage of the KB, sentences with missing annotations are contained in the corpus. For example, although the tokens *Chateau* and *Talbot* in Figure 4 should be annotated as B-PRODUCER and E-PRODUCER<sup>5</sup> respectively, they are not annotated. These missing annotations result in reduced performance (especially *recall*) of models based on the corpus since they are considered to be *other* examples when training the models. One of the possible way to reduce the influence of the missing annotations is to discard sentences with missing annotations. Thus, we removed such sentences from the corpus.

To detect missing annotations, we generate morpheme patterns from values in the KB. Ex-

<sup>5</sup>The beginning and the end of a value for the attribute called “Producer.”

Table 2: Features for training extraction models.

Basic features (BFs)	
Feature	Description
Token	Surface form of the token.
Base	Base form of the token.
PoS	Part-of-speech tag of the token.
Char. types	Types of characters contained in the token. We defined <i>Hiragana</i> , <i>Katakana</i> , <i>Kanji</i> , <i>Alphabet</i> , <i>Number</i> , and <i>Other</i> .
Prefix	Double character prefix of the token.
Suffix	Double character suffix of the token.
Context features	
Feature	Description
Context	BFs of $\pm 3$ tokens surrounding the token.

amples of the patterns for the wine category are given in Figure 5. First, attribute values are tokenized using a morphological analyzer, and then the PrefixSpan algorithm (Pei et al., 2001) is executed on the tokenized result to induce morpheme patterns<sup>6</sup>. We employed patterns that do not start and end with the [ANY] token, and which match attribute values in the KB and the total number of merchants corresponding to the matched values is greater than one.

### 4.3 Extraction Model Training

Models for attribute value extraction are trained using the corpus. We employed Conditional Random Fields (CRFs), and used CRFsuite<sup>7</sup> with default parameter settings as the implementation thereof. The features we used are listed in Table 2.

We built a single model for each category. In other words, we did not build a separate model for each attribute of each category.

## 5 Experiments

This section reports the experimental results. We carried out the experiments on eight categories, and evaluated them using the dataset discussed in Section 5.1. An evaluation was conducted for each component, that is, evaluation of the KB (Section 5.2), evaluation of the automatically annotated corpora (Section 5.3), and evaluation of the extraction models (Section 5.4).

### 5.1 Construction of Evaluation Dataset

We created an evaluation dataset comprising 1,776 product pages gathered from the eight categories in Table 3. In constructing the dataset, for each category, we first listed top 300 merchants according to the number of products offered by the mer-

<sup>6</sup>We used PrefixSpan-rel as the implementation of PrefixSpan. (<http://prefixspan-rel.sourceforge.jp/>)

<sup>7</sup><http://www.chokkan.org/software/crfsuite/>

Table 3: Categories and their selected attributes. The symbol # denotes incorrect attributes. The symbol \* is attached to the attributes that are not aggregated into their synonyms. For example, “country of origin” for wine is marked because it is not aggregated with “production area”.

Category	Attributes (Each attribute is represented by one of its synonyms.)
Wine	容量 (volume), 品種 (grape variety), タイプ (type), 産地 (production area), アルコール度数 (alcohol), 原産国 (country of origin)*, 生産者 (producer), 原材料 (material)
T-shirt(men)	サイズ (size), 素材 (material), 色 (color), 着丈 (length)*, 身幅 (body width)*, M(M size)#, 肩幅 (shoulder width)*, L(L size)#
Printer ink	容量 (volume), サイズ (size), カラー (color), 重量 (weight), 色 (color)*, 適応機種 (compatible model), 材質 (material), 製造国 (production area)
Shampoo	容量 (volume), メーカー (manufacturer), 製造国 (production area), 成分 (constituent), 商品名 (product name), 区分 (category), サイズ (size), 重量 (weight)
Golf ball	コア (core), サイズ (size), カバー (cover), 材質 (material), 重さ (weight), 原産国 (country of origin), デインプル形状 (shape of dimple), 色 (color)
Video game	サイズ (size), 重さ (weight), 材質 (material), 付属品 (accessory), 製造国 (production country), 色 (color), 対応機種 (compatible model), ケーブル長 (length of cable)
Car spotlight	サイズ (size), 色温度 (color temperature), 色 (color), 材質 (material), 重量 (weight), 適合車種 (compatible model), 製造国 (production country), 品番 (part number)
Cat food	内容量 (volume), 原産国 (production country), 粗繊維 (crude fiber), 粗脂肪 (crude fat), 粗灰分 (crude ash), 水分 (wet), 粗タンパク質 (crude protein), 重量 (weight)*

Table 4: Statistics of the corpora.  $p^{\#}$ ,  $s^{\#}$ , and  $v^{\#}$  denote the number of annotated pages, the number of annotated sentences, and the number of values, respectively.

Category	Test data (manually annotated)			Training data (automatically annotated)		
	$p^{\#}$	$s^{\#}$	$v^{\#}$	$p^{\#}$	$s^{\#}$	$v^{\#}$
Wine	282	1,863	3,040	25,358	28,952	48,645
T-shirt(men)	259	2,580	5,534	14,978	18,018	41,954
Printer ink	273	1,230	4,029	8,473	13,562	42,969
Shampoo	233	1,518	4,352	18,669	30,263	53,294
Golf ball	160	555	719	1,114	2,109	2,760
Video game	212	807	1,088	19,292	29,356	35,230
Car spotlight	271	1,401	2,282	8,124	12,910	18,937
Cat food	86	276	452	4,915	7,375	8,843
Total	1,776	10,230	19,496	100,923	142,545	252,632

chant in the category. Then, we randomly picked one from the product pages of each merchant. We extracted titles and sentences from the pages based on HTML tags. These texts were passed to the annotation process.

In selecting the attributes to be used for annotation, we selected the top eight attributes in each category according to the MFs of the attributes. Then, we manually discarded incorrect attributes and aggregated synonymous attributes that were not automatically discovered. These attributes are marked up with the symbols # and \* in Table 3, and are not considered in the evaluation in Sections 5.3 and 5.4.

An annotator was asked to annotate expressions in the text data, which could be considered as values of the selected attributes. The annotator was also asked to discard pages that offered multiple products and miscategorized products.

In this way, the evaluation dataset was constructed by one annotator. After the construction,

we checked the accuracy of the annotation. We picked up 400 annotated pages, and then checked them by another annotator whether annotations in the pages were correct. The agreement of annotations between the two annotators was about 88.4%. Statistics of the dataset are given in the *Test data* column in Table 4.

## 5.2 Evaluation of Knowledge Base

### 5.2.1 Evaluation of Extracted Attributes

We checked whether attributes extracted using our method could be regarded as *correct*. We asked two subjects to judge 411 expressions, all extracted attributes for the eight categories. The ratio of attributes that were judged as *correct* by both annotators was 0.776. The kappa statistics between the annotators was 0.581. This value is defined as *moderate agreement* in (Landis and Koch, 1977). Majority of the attributes judged as *incorrect* were extracted from complex tables and tables on miscategorized pages.

### 5.2.2 Evaluation of Attribute Synonyms

Next, we assessed the performance of our synonym discovery. We asked the subjects to aggregate synonymous attributes in KBs, and then computed *purity* and *inverse purity* scores (Artiles et al., 2007) using the data. We discarded attributes judged as incorrect when computing these scores. We computed macro-averaged scores for each subject, and then averaged them.

As a result, the averaged purity and inverse purity were 0.920 and 0.813, respectively. The purity score is close to perfect, which means that the merged expressions are mostly regarded as synonyms. On the other hand, the score for in-

Table 5: Accuracy of KBs. # shows the total number of KB entries with each MF.

MF of pairs	Wine		Shampoo	
	#	Acc. [%]	#	Acc. [%]
≥ 1 (All)	3,940	75.3 (301/400)	6,798	67.5 (270/400)
≥ 2	751	97.2 (69/71)	2,307	90.8 (118/130)
≥ 3	384	97.1 (33/34)	1,543	97.3 (73/75)
≥ 5	215	95.5 (21/22)	931	98.1 (52/53)

Table 6: Accuracy of our annotation method.

Annotation method	Prec. (%)	Recall (%)	F <sub>1</sub> score
(1) Naive annotation	47.46	<b>45.48</b>	<b>46.19</b>
(2) (1) + incorrect	51.39	39.14	43.00
(3) (2) + missing	<b>57.14</b>	29.29	37.21

verse purity is less than 0.82. Improvement of the methodology in terms of coverage is left for future work.

### 5.2.3 Evaluation of Attribute Values

We evaluated the quality of the KBs for the wine and shampoo categories only, because the evaluation for all categories requires enormous human effort. We randomly selected 400 values of attributes listed in Table 3, and then asked the subjects to judge whether the values could be regarded as *correct* for the attributes. To judge the pair  $\langle attr., value \rangle$  in the KB for category  $C$ , we automatically generated the sentence: “ $value$  is an expression that represents ( $attr.$  or  $S_{attr}^1$  or ... or  $S_{attr}^n$ ) of  $C$ .” Here,  $S_{attr}^n$  denotes the  $n$ th synonym of  $attr.$  The subjects judged a pair to be *correct* if the sentence generated from the pair was naturally acceptable. For example, the pair  $\langle variety, onion \rangle$  in the KB for the wine category is deemed incorrect because the sentence “*onion* is an expression that represents (*variety* or *grape variety* or *usage variety*) of *wine*.” is not acceptable.

The evaluation results are given in Table 5. The kappa statistics between the annotators were 0.632 for the wine category, and 0.678 for the shampoo category, respectively. These values indicate *good agreement*. We regarded a pair as correct if the pair was judged as correct by both annotators. We can see that the accuracy of each category is promising. In particular, the accuracy of pairs with MF greater than one is 90% or more. This means that merchant frequency plays a crucial role in constructing KBs from structured data that are embedded in product pages by different merchants.

### 5.3 Evaluation of the Annotated Corpora

We also checked the effectiveness of the proposed annotation method by annotating the same product

<b>KB match:</b> Naive KB matching for corpora (same as (3) in Table 6).
<b>Model w/o filters:</b> Training models based on corpora naively annotated using KBs. That is, filters for incorrect and missing annotations are not applied.
<b>Model with incorrect annotation filter (Model with incorrect only):</b> Training models based on corpora where only the filter for incorrect annotations is applied.
<b>Model with missing annotation filter (Model with miss only):</b> Training models based on corpora where only the filter for missing annotations is applied.

Figure 6: Alternative methods.

Table 7: Micro-averaged precision, recall and F score of the models for proposed method and alternatives.

Method	Prec.(%)	Recall (%)	F <sub>1</sub> score
Supervised Model	88.28	58.15	68.66
KB match	57.14	29.29	37.21
Model w/o filters	52.60	54.49	53.14
Model with incorrect only	<b>60.46</b>	54.23	56.84
Model with miss only	50.47	<b>59.71</b>	54.43
Model for proposed method	57.05	59.66	<b>58.15</b>

pages as those in the evaluation dataset, and then checking overlaps between the manual and automatic annotations. The experimental results are given in Table 6. We judged an extracted value to be *correct* if the value exactly matched the manually annotated one. The results are given as micro-averaged precision, recall, and F<sub>1</sub> scores for each attribute in each category. We can see that the proposed filtering methods improve the precision (annotation quality) at the expense of recall.

### 5.4 Evaluation of Extraction Model

We compared the performance of the extraction models trained for each category with the alternative methods shown in Figure 6. We naively matched entries in the KB for unlabeled product pages, and then randomly picked 100,000 unique sentences from the annotated pages. We refer to the picked sentences as the *Raw Corpus* (RC). Then, we ran the filters and training process on the RC since we were limited by the RAM required by CRFSuite. Some statistics of the corpora after applying the filters are shown in the column *Training data* in Table 4.

The evaluation results are shown in Table 7. **Model w/o filters** outperformed **KB match** by as much as 15.9 points in F<sub>1</sub> score. These improvements are caused by improving the recall of the method. This shows that contexts surrounding a value and patterns of tokens in a value are successfully captured. **Model with incorrect only** also achieved higher performance than **Model w/o**

Table 8: Types of errors.

Type		# err.
Automatic annotation	Context dependent role	15
	Part of a compound word	12
	Polysemous word	9
Incorrect KB entry		23
Over generation by	learned patterns	15
Extraction from	unrelated regions	12
Others		14

**filters.** Especially, the precision of the extraction models is improved by 7.9%. This means that the incorrect annotation filter successfully removed annotation based on incorrect KB entries from the annotated corpora. In addition, **Model with miss only** achieved higher performance than **Model w/o filters**. In particular, the recall of the method improved by 5.2%. This shows that the missing annotation filter effectively works for precisely training extraction models. As a result, the precision and recall of the proposed method are enhanced by employing both filters simultaneously, and the method achieved 58.15 points in F<sub>1</sub> scores.

By comparison, the performance of the extraction models based on manually annotated corpora is shown in Table 7. The supervised method was evaluated with 10-fold cross validation. From the table, we can see that the recall of our method outperforms that of the supervised method while the precision and F<sub>1</sub> score of our method are lower those of the supervised method.

## 6 Discussion

For the wine and shampoo categories, we randomly picked up 50 attribute values that were judged as incorrect. Then we classified them according to their error types. The classification result is shown in Table 8. Ratios of errors based on automatic annotation, and incorrect KB entities, over generation by learned patterns, and extraction from unrelated region with products are 36%, 23%, 15% and 12%, respectively.

The errors stemming from automatic annotation can be classified into three sub-types. Errors that require understanding of the context when annotating attribute values are the most common sub-type. For example, in the wine domain, the attribute-value pair <Production area, Bordeaux> was extracted from the following sentence:

- 土壌が ボルドー<Productionarea> のポムロールと非常に似ている。  
(Soil is very similar with ones in Pomerol region of Bordeaux<Productionarea>.)

Although the extracted pair can be regarded as a correct KB entity for the wine categories, it is not *production areas* of wine in the above sentence. This type of error can be reduced if we can successfully leverage the above sentence as a negative example in the model training step. To generate such negative examples is future work.

The second type of errors left for future work occurs in annotation of compound words. For example, automatically annotated corpora for the shampoo category has the following sentence:

- ヒアルロン酸<Constituent> 以上の保水力がある。  
(It has a higher water-holding ability than hyaluronan<Constituent> has.)

This sort of annotation errors may decrease if we omit the annotation of parts of compound words.

The third type of errors is annotation based on polysemous words. For instance, although <Alcohol, 10%> for the wine category is a correct KB entry, the word “10%” is used for describing various types of ratios. The following sentence is one of the examples where the word 10% is used with a meaning other than alcohol content in the wine domain:

- 輸出は全体の 10%<Alcohol> 程度。  
(The amount of exports is approximately 10%<Alcohol> of the total.)

A wrong extraction model for the alcohol attribute is trained through the above sentence. Disambiguation of attribute values is required in the annotation step in order to train precise models. On the task of extracting person names, a method for disambiguation of names is proposed by Bollegala et al. (2012). To employ similar disambiguation methodology is one of our plans for future work.

## 7 Conclusion

We proposed a purely unsupervised methodology for extracting attributes and their values from e-commerce product pages. We showed that the performance of our method attained an average F score of approximately 58.2 points using manually annotated corpora.

We believe the most important task for future work is to improve annotation quality. Disambiguation of attribute values and construction of wide coverage KBs are crucial to boost the quality. Another important future task concerns synonymy. We only tackled attribute synonymy. Discovery of attribute value synonyms is also an important future direction.

## References

- Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2007. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 64–69.
- Lidong Bing, Tak-Lam Wong, and Wai Lam. 2012. Unsupervised extraction of popular product attributes from web sites. In *Proceedings of the Eighth Asia Information Retrieval Societies Conference*, pages 437–446.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2012. Automatic annotation of ambiguous personal names on the web. *Computational Intelligence*, 28(3):398–425.
- Nilesh N. Dalvi, Philip Bohannon, and Fei Sha. 2009. Robust web extraction: an approach based on a probabilistic tree-edit model. In *Proceedings of the 2009 ACM International Conference on Management of Data*, pages 335–348.
- Remi Ferrez, Clement Groc, and Javier Couto. 2013. Mining product features from the web: A self-supervised approach. In *Web Information Systems and Technologies*, volume 140 of *Lecture Notes in Business Information Processing*, pages 296–311.
- Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48.
- Pankaj Gulhane, Rajeev Rastogi, Srinivasan H. Sengamedu, and Ashwin Tengli. 2010. Exploiting content redundancy for web information extraction. In *Proceedings of the 19th International World Wide Web Conference*, pages 1105–1106.
- Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 407–415.
- J. Richard Landis and Gary. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.
- Karin Mauge, Khash Rohanimanesh, and Jean-David Ruvini. 2012. Structuring e-commerce inventory. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 805–814.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the Fourth International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.
- Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining*, pages 577–583.
- Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th IEEE International Conference of Data Engineering*, pages 215–224.
- Katharina Probst, Rayid Ghani, Marko Krema, Andrew Fano, and Yan Liu. 2007. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of the 20th International Joint Conference in Artificial Intelligence*, pages 2838–2843.
- Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567.
- Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 104–111.
- Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinou. 1998. A decision tree method for finding and classifying names in japanese texts. In *In Proceedings of the Sixth Workshop on Very Large Corpora*.
- Keiji Shinzato and Kentaro Torisawa. 2004. Acquiring hyponymy relations from web documents. In *Proceedings of Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting*, pages 73–80.
- Kentaro Torisawa. 2001. An unsupervised method for canonicalization of japanese postpositions. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, pages 211–218.
- Tak-Lam Wong, Wai Lam, and Tik-Shun Wong. 2008. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the 31st ACM SIGIR Conference*, pages 35–42.
- Naoki Yoshinaga and Kentaro Torisawa. 2006. Finding specification pages according to attributes. In *Proceedings of the 15th International World Wide Web Conference*, pages 1021–1022.



# Stance Classification of Ideological Debates: Data, Models, Features, and Constraints

Kazi Saidul Hasan and Vincent Ng

Human Language Technology Research Institute

University of Texas at Dallas

Richardson, TX 75083-0688

{saidul, vince}@hlt.utdallas.edu

## Abstract

Determining the stance expressed in a post written for a two-sided debate in an online debate forum is a relatively new and challenging problem in opinion mining. We seek to gain a better understanding of how to improve machine learning approaches to stance classification of ideological debates, specifically by examining how the performance of a learning-based stance classification system varies with the amount and quality of the training data, the complexity of the underlying model, the richness of the feature set, as well as the application of extra-linguistic constraints.

## 1 Introduction

Determining the stance expressed in a post written for a two-sided debate in an online debate forum is a relatively new task in opinion mining. Given a post written for a *two-sided* topic discussed in an online debate forum (e.g., “*Should abortion be banned?*”), the goal of *debate stance classification* is to determine which of the two sides (i.e., *for* and *against*) its author is taking.

Previous approaches to debate stance classification have focused on three debate settings, namely congressional floor debates (e.g., Thomas et al. (2006), Bansal et al. (2008), Balahur et al. (2009), Yessenalina et al. (2010), Burfoot et al. (2011)), company-internal discussions (e.g., Agrawal et al. (2003), Murakami and Raymond (2010)), and online social, political, and ideological debates in public forums (e.g., Somasundaran and Wiebe (2010), Wang and Rosé (2010), Anand et al. (2011), Biran and Rambow (2011), Hasan and Ng (2012)). As Walker et al. (2012) point out, debates in public forums differ from congressional debates and company-internal discussions in terms of language use. Specifically, online debaters use colorful and emotional language to express their points,

which may involve sarcasm, insults, and questioning another debater’s assumptions and evidence. These properties could make stance classification of online debates more challenging than that of the other two types of debates.

Our goal in this paper is to gain a better understanding of how to improve machine learning approaches to stance classification of online debates by examining the following questions, which can be broadly categorized along four dimensions:

**Data.** Can we improve the performance of a stance classification system simply by increasing the number of stance-annotated debate posts available for training? Note, however, that the number of stance-annotated posts that can be downloaded from debate forums for a given debate domain (e.g., Abortion) is fairly limited. A natural question is: given a debate domain, can we identify from different data sources a large number of documents where authors express viewpoints relevant to the domain (e.g., blog posts, news articles) and then stance-label them heuristically, with the goal of employing these noisily labeled documents as additional data for training a stance classifier?

**Features.** The simplest kind of features one can think of is probably n-grams. Nevertheless, a stance classifier trained on unigrams is a relatively strong baseline (Somasundaran and Wiebe, 2010). Anand et al. (2011) augment an n-gram feature set with four types of features: document statistics, punctuations, syntactic dependencies, and, if applicable, the set of features computed for the immediately preceding post in the discussion thread (see Section 3 for details). How effective are Anand et al.’s features in improving an n-gram-based stance classifier? Will adding semantic features improve performance further?

**Models.** The simplest stance classification model is probably one that assigns a stance label to each debate post independently of the other posts. Can we get better performance by

exploiting the linear structure inherent in a post sequence? Since a post may contain materials irrelevant to stance classification, can we train a better model by learning only from the stance-related sentences without relying on sentences manually annotated with stance labels?

**Constraints.** Extra-linguistic inter-post constraints, such as *author constraints* (see Section 3), have been shown to be effective in improving stance classification performance by postprocessing the output of a stance classifier. Will the effectiveness of these constraints be dependent on the underlying debate domain? Will it be dependent on the accuracy of the stance classifier to which they are applied?

By examining these questions, we can potentially determine how the performance of a stance classification system varies with the amount and quality of the training data, the complexity of the underlying model, the richness of the feature set, as well as the application of extra-linguistic constraints. In our evaluation, we focus on stance classification of *ideological debates*.

## 2 Datasets

For our experiments, we collect debate posts from four popular *domains*, Abortion (ABO), Gay Rights (GAY), Obama (OBA), and Marijuana (MAR). Each post should receive one of two *domain labels*, *for* and *against*, depending on whether the author of the post *supports* or *opposes* abortion, gay rights, Obama, and the legalization of marijuana, respectively. To see how we obtain these domain labels, let us first describe the data collection process in more detail.

We collect our debate posts for the four domains from an online debate forum<sup>1</sup>. In each domain, there are several two-sided debates. Each debate has a subject (e.g., “Abortion should be banned”) for which a number of posts were written by different authors. Each post is manually tagged with its author’s stance (i.e., *yes* or *no*) on the debate subject. Since the label of each post represents the subject stance but not the domain stance, we need to automatically convert the former to the latter. For example, for the subject “Abortion should be banned”, the subject stance *yes* implies that the author opposes to abortion, and hence the domain label for the corresponding label should be *against*.

<sup>1</sup><http://www.createdebate.com/>

Domain	Posts	% of “for” posts	% posts in a thread	Average thread length
ABO	1741	54.9	75.1	4.1
GAY	1376	63.4	74.5	4.0
OBA	985	53.9	57.1	2.6
MAR	626	69.5	58.0	2.5

Table 1: Statistics of the four datasets.

We construct one dataset for each domain. Statistics of these datasets are shown in Table 1.

## 3 Experimental Setup

In this section, we describe the experimental setup behind our investigation of the issues along the four dimensions of learning-based stance classification: models, features, data, and constraints.

### 3.1 Models

We seek to examine how model *complexity* impacts stance classification performance. We consider three types of models.

The first type of models is a binary classifier that assigns a stance label to each debate post independently of the other posts. We employ a generative model (Naive Bayes (NB) with add-one smoothing) and a discriminative model (Support Vector Machines (SVMs), as implemented in SVM<sup>light</sup> (Joachims, 1999)) in our investigation. This enables us to determine whether the relative performance of generative models and discriminative models changes with the amount of training data (Ng and Jordan, 2002), and whether generative models can handle complex, possibly overlapping features as well as discriminative models.

The second type of models, sequence models, is motivated by an observation: since a post in a post sequence is a reply to its parent post, its label should be determined in dependent relation to that of its parent. Consequently, these models assume as input a post sequence and output a sequence of stance labels, one for each post in the input sequence. As before, we employ two sequence labelers, one generative (first-order Hidden Markov Models (HMMs) with add-one smoothing) and one discriminative (linear-chain Conditional Random Fields (CRFs) (Lafferty et al., 2001), as implemented in Mallet (McCallum, 2002)).

The last type of models is *fine-grained* models. These models jointly determine the stance label of a debate post and the stance label of each of its sentences. We hypothesize that modeling sentence

stances could improve document stance classification performance: for example, features computed from sentences with a neutral stance should not play any role in determining the document stance. To avoid the cost of hand-annotating sentences with stance labels for training a sentence-stance classifier, we determine sentence stances in our model in an unsupervised manner. Moreover, while it is possible to implement fine-grained models based on NB, SVM, HMM, and CRF, we will focus exclusively on those based on NB and HMM. The reason is that they are easier to implement because we employ our own implementation of NB and HMM in these experiments.

**The generative story.** To generate a document  $d_i$ , we first pick a document stance  $c$  with probability  $P(c)$ . Given  $c$ , we generate each sentence in  $d_i$  independently of each other. To generate a sentence  $e_m$ , we first pick a sentence stance  $s$  with probability  $P(s|c)$ , and then generate  $f_n$ , the value of the  $n$ th feature representing  $e_m$ , with probability  $P(f_n|s, c)$ .

A few points deserve mention. First, fine-grained NB and fine-grained HMM both employ this document generative story, differing only in terms of whether the document stance is generated independently (NB) or in dependent relation to that of the preceding post (HMM). Second, while a document stance can have one of two possible values (*for* and *against*), a sentence stance can have one of three possible values (*for*, *against*, and *neutral*). Note that the stance of a sentence can be expressed by someone other than the author of the post; for example, a sentence may restate an opposing opinion by a different author.

**Training the fine-grained models.** As noted above, we need to estimate  $P(c)$ ,  $P(s|c)$ , and  $P(f_n|s, c)$ .  $P(c)$  can be estimated from the stance-labeled training documents.<sup>2</sup> However, since sentence stances are hidden, we estimate  $P(s|c)$  and  $P(f_n|s, c)$  using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

To employ EM, we begin by heuristically labeling each sentence with a stance as follows. First, for each document stance  $c$ , we identify the list of informative unigrams, which consists of the open-class words that appear at least 10 times in the training data and is associated with  $c$  at least 70%

<sup>2</sup>In the case of HMMs, we need to additionally estimate the document transition probabilities, which can be done in a supervised manner.

of the times. Then, given a sentence  $e_m$ , its stance label is determined by taking a simple majority vote using the stance labels associated with the informative unigrams appearing in  $e_m$ . In case of a tie,  $e_m$  is labeled as *neutral*.<sup>3</sup>

After heuristic labeling, we begin with the M-step, where we estimate model parameters  $P(s|c)$  and  $P(f_n|c, s)$  from the training data. Since the data is now stance-labeled at both the document and sentence level, we can estimate these parameters using maximum likelihood estimation.

Next, we proceed to the E-step, where the goal is to estimate  $P(s|e_m, d_i, c)$ , the probability that a sentence expresses a sentence stance given the document stance. From the above generative story,

$$\begin{aligned} P(s|e_m, d_i, c) &\propto P(s|c)P(e_m, d_i|s, c) \\ &= P(s|c) \prod_{n=1}^{|F|} P(f_n|s, c) \end{aligned} \quad (1)$$

where  $F$  is the set of features in sentence  $e_m$ . We run EM until convergence.

**Applying the fine-grained models.** After training, we can apply the fine-grained models to classify each test post  $d_i$ . For fine-grained NB, we employ the following equation:

$$\begin{aligned} P(c|d_i) &\propto P(c)P(d_i|c) \\ &= P(c) \prod_{m=1}^{|S(d_i)|} P(s_{max}|e_m, d_i, c) \end{aligned} \quad (2)$$

where  $S(d_i)$  is the set of sentences in test post  $d_i$ , and  $s_{max}$  is the sentence stance with the maximum conditional probability (obtained using Equation 1) for sentence  $e_m$  in  $d_i$ .

For fine-grained HMM, we employ Viterbi to decode a post sequence, using  $P(d_i|c)$  as the “output probability” of a test post given stance  $c$ .

### 3.2 Features

We seek to examine how the features used to train the stance classification system impact its performance. We consider three feature sets.

**N-gram features.** The first feature set consists of unigrams and bigrams collected from the training posts. We encode them as binary features that indicate their presence or absence in a given post.

<sup>3</sup>Intuitively, sentences containing an equal number of *for* and *against* cues are not neutral. We label them as neutral simply because there is no reason to prefer one non-neutral stance to another.

Sentence: Every woman has the right to choose abortion.		
Frame	Target/Semantic Role of Element	Text
People	Target	woman
Possession	Target	has
	Owner	Every woman
	Possession	the right to choose abortion
Correctness	Target	right
Choosing	Target	choose
	Chosen	abortion

Table 2: Sample frame-semantic parse.

**Anand et al.’s (2011) features.** The second feature set, proposed by Anand et al., consists of five types of features: n-grams, document statistics, punctuations, syntactic dependencies, and, if applicable, the set of features computed for the immediately preceding post in its thread. Their n-gram features include both the unigrams and bigrams in a post, as well as its first unigram, first bigram, and first trigram. The features based on document statistics include the post length, the number of words per sentence, the percentage of words with more than six letters, and the percentage of words as pronouns and sentiment words. The punctuation features are composed of the repeated punctuation symbols in a post. The dependency-based features have three variants. In the first variant, the pair of arguments involved in each dependency relation extracted by a dependency parser is used as a feature. The second variant is the same as the first except that the head (i.e., the first argument in a relation) is replaced by its part-of-speech (POS) tag. The features in the third variant, the topic-opinion features, are created by replacing each feature from the first two types that contains a sentiment word with the corresponding polarity label (i.e., + or -).

**Adding frame-semantic features.** To provide semantic generalizations, we create features computed using FrameNet semantic frames (Baker et al., 1998). More specifically, we first apply SEMAFOR (Das et al., 2010) to create a frame-semantic parse for each sentence in a given debate post. Then, for each frame that a sentence contains, we create three types of frame-semantic features, as described below.

A *frame-word interaction feature* is a binary feature composed of (1) the name of the frame  $f$  from which it is created, and (2) an unordered word pair in which the words are taken from two frame elements of  $f$ . Specifically, for each pair of frame elements  $fe_1$  and  $fe_2$  of a frame  $f$ , we cre-

ate one frame-word interaction feature from each unordered word pair composed of one word from  $fe_1$  and one word from  $fe_2$ . Consider the frame-semantic parse of the sentence *Every woman has the right to choose abortion* shown in Table 2. Given the frame *Possession* and its frame elements *Every woman* and *the right to choose abortion*, we can generate frame-word interaction features such as *Possession-right-woman*, *Possession-choose-woman*, *Possession-abortion-woman*.

A *frame-pair feature* is a binary feature composed of a word pair corresponding to the names of two frames, in which the target of the first is present in a frame element of the second. Specifically, for each frame element  $fe$  of a frame  $f$ , if a substring of  $fe$  is the target of a frame  $f_2$ , we create a frame-pair feature composed of the ordered pair  $(f_2, f)$ . Consider the example in Table 2 again. Given the frame *Possession* and its frame elements *Every woman* and *the right to choose abortion*, we can create three frame-pair features, *People:Possession*, *Choosing:Possession*, and *Correctness:Possession*, since *woman*, *choose*, and *right* are the targets of frames *People*, *Choosing*, and *Correctness*, respectively.

A *frame n-gram feature* is the frame-based version of a word n-gram feature. Given a word unigram or bigram in which each word is an open-class word, we create all possible frame n-gram features from it by replacing one or more of its words with its frame name (if the word is a frame target) or its frame semantic role (if the word is present in a frame element). For instance, in the word bigram *woman+has* from the sentence in Table 2, both *woman* and *has* are open-class words and are targets of *People* and *Possession*, respectively. Hence, we create for *woman+has* three frame n-gram features: *woman+Possession*, *People+has*, and *People+Possession*. In addition, since *woman* plays the role of *Owner* in *Possession*, we create two more frame n-gram features, *Owner+Possession* and *Owner+has*.

**Using the frame-semantic features.** One way to use the frame-semantic features is to incorporate them into Anand et al.’s (2011) feature set and train a stance classifier on the augmented feature set.<sup>4</sup> We employ a different way of using the frame-semantic features, however. We train two

<sup>4</sup>Preliminary results indicate that training a stance classifier on the augmented feature set does not yield good performance, presumably because the frame-semantic features are significantly outnumbered by Anand et al.’s features.

Abortion		Gay Rights	
<i>For</i>	<i>Against</i>	<i>For</i>	<i>Against</i>
I think abortion should be legal.	I think abortion should not be legal.	I support gay marriage.	I do not support gay marriage.
I support abortion.	I do not support abortion.	I support gay adoption.	I do not support gay adoption.
I think abortion should be allowed.	I think abortion should not be allowed.	I am in favor of same-sex marriage.	I am against same-sex marriage.
I think abortion should not be banned.	I think abortion should be banned.	I think gay marriage should be legal.	I think gay marriage should not be legal.
Obama		Marijuana	
<i>For</i>	<i>Against</i>	<i>For</i>	<i>Against</i>
I support President Obama.	I do not support Obama.	I think marijuana should be legalized.	I think marijuana should not be legalized.
I am a fan of Barack Obama.	I am against Obama.	I think marijuana should not be banned.	I think marijuana should be banned.
I like President Obama.	I do not like Obama.	I support marijuana legalization.	I do not support marijuana legalization.
I will vote for Obama.	I will not vote for Obama.	I think marijuana should not be illegal.	I think marijuana should be illegal.

Table 3: Sample search queries.

stance classifiers,  $C_A$  and  $C_{FS}$ .  $C_A$  is trained using Anand et al.’s (2011) features, whereas  $C_{FS}$  is trained using only the frame-semantic features. After training, we use the classifiers to predict the stance for a post  $x$  in the test set as follows. We first apply them independently to classify  $x$ , and then predict the stance for  $x$  by linearly interpolating the resulting classification values. The value of the interpolation constant is tuned to maximize performance on development data.<sup>5</sup>

### 3.3 Data

We seek to examine how the *amount* and *quality* of the training data impact stance classification performance.

To determine how classification performance varies with the amount of training data, we will plot learning curves in our evaluation.

As far as training data quality is concerned, our goal is to collect documents discussing viewpoints relevant to the debate domain of interest from different sources (e.g., blogs, news websites), stance-label them heuristically, and determine how these noisily labeled documents can be used in combination with the stance-annotated debate posts to train a stance classification system. Below we describe how we collect and utilize these documents.

**Collecting noisily labeled documents.** To collect noisily labeled documents, we employ a two-step procedure. We (1) create using commonsense knowledge a list of phrases that are reliable indicators of both stances for each domain; and then (2) use each phrase as an *exact* search query to retrieve noisily labeled documents from the Web.

Sample phrases that we create for each stance of each domain are shown in Table 3.<sup>6</sup> For instance, for the Abortion domain, the phrase *I support abortion* indicates the author’s support for

abortion. In contrast, *I think abortion should be banned* is indicative of the author’s stance against abortion. Since we use each phrase as a search query in the second step, we manually paraphrase each of them in hope to increase the number of retrieved documents. For instance, we create for *abortion should be banned* paraphrases such as *abortion should be prohibited*, *abortion should be illegal*, and *abortion should not be allowed*. Some paraphrases are created simply by employing different forms of a proper noun (e.g., *Obama*, *Barack Obama*, and *President Obama*). Table 4 shows the statistics of the noisily labeled documents. It took us less than two person-days to create the phrases and their paraphrases for each domain. Roughly the same number of phrases were created for the two stances in a domain.

As noted above, we use each phrase created in the first step as an exact search query to retrieve documents from the Web using Bing’s Search API.<sup>7</sup> A closer inspection of the retrieved documents reveals that many of them contain materials irrelevant to the search query. One of them, for instance, is a blog article discussing different facets of women rights, followed by comments from several readers. The search query that retrieved the document appeared in one of the readers’ comments. In this case, it makes sense to delete everything but this reader’s comment from the document before using it as noisily labeled data. For this reason, we heuristically extract the portion of each retrieved document that is relevant to the search query. More specifically, we define the relevant portion of a document as the smallest string that contains the search query string and is delimited by HTML tags. Note that we discard documents that contain less than 10 words (in order to avoid documents with no useful content) or are retrieved from `www.createdebate.com` (in or-

<sup>5</sup>We tried values from 0.0 to 1.0 in steps of 0.001.

<sup>6</sup>The complete set of phrases is available at <http://www.hlt.utdallas.edu/~saidul/stance.html>.

<sup>7</sup><https://datamarket.azure.com/dataset/bing/search>

Domain	Phrases	Posts	% of “for” posts
ABO	125	10187	43.6
GAY	438	8148	62.5
OBA	205	9687	54.1
MAR	376	3333	57.7

Table 4: Noisy data statistics.

der to avoid overlaps with our evaluation datasets).

**Training with noisily labeled documents.** Given these noisily labeled documents, how can they be used in combination with the (cleanly labeled) debate posts in the training set for training stance classifiers? Motivated by Nguyen and Moschitti (2011), we train two stance classifiers,  $C_c$  and  $C_{c+n}$ .  $C_c$  is trained on only the debate posts in the training set.  $C_{c+n}$ , on the other hand, is trained on both the debate posts and the noisily labeled documents.<sup>8</sup> Both of them use the same set of features.

After training, we use these classifiers to predict the stance for a post  $x$  in the test set as follows. We first apply them independently to classify  $x$ , and then predict the stance for  $x$  by linearly interpolating the resulting classification values. The value of the interpolation constant is tuned to maximize performance on development data.<sup>9</sup>

### 3.4 Constraints

Previous work on stance classification of congressional debates has found that enforcing author constraints (ACs) can improve classification performance (e.g., Thomas et al. (2006)). ACs are a type of inter-post constraints that specify that two posts written by the same author for the same debate domain should have the same stance, and are typically used to postprocess the output of a stance classifier. We seek to determine how ACs impact the performance of a system for stance-classifying ideological debate posts, and whether their effectiveness depends on the debate domain.

In our experiments, we enforce ACs as follows. We first use a stance classifier to classify the test posts. Note that the classification value of a post can be thought of as a probabilistic vote that a post can cast on the stance labels. Then, given a set of

<sup>8</sup>We treat the noisily labeled documents as sequences of length one when using them to train HMMs and CRFs.

<sup>9</sup>We tried values from 0.0 to 1.0 in steps of 0.001. Note that when both frame-semantic features and noisily labeled documents are used, there are two interpolation constants to be tuned. In that case, we tune the constant associated with the frame-semantic features before tuning the one associated with the noisily labeled documents.

test posts written by the same author for the same debate domain, we sum up the probabilistic votes cast by these posts, and assign to each of them the stance that receives the larger number of votes.

## 4 Evaluation

In the previous section, we described the experimental setup for investigating the issues pertaining to the four dimensions of learning-based stance classification. In this section, we begin by describing the general experimental setup and then report on and discuss the evaluation results.

### 4.1 General Experimental Setup

Results are expressed in terms of *accuracy* obtained via 5-fold cross validation, where accuracy is the percentage of test instances correctly classified. Since all experiments require the use of development data for parameter tuning, we use three folds for model training, one fold for development, and one fold for testing in each fold experiment. All SVM and CRF learning parameters are set to their default values in SVM<sup>light</sup> and Mallet, respectively.

Learning curves are generated for all the experiments. Each point on a learning curve is computed by averaging the results of five independent runs corresponding to five different randomly selected training sets of the required size. To ensure a fair comparison of different learning models, the same five randomly selected training sets of the required size are used to train the models. Since the models based on HMMs and CRFs need to be trained on post sequences, we assemble a training set of a given size as follows: whenever a post is sampled for inclusion into the training set, we incorporate all the posts in the same post sequence into the training set.

### 4.2 Results

Results for the four domains are shown as four sub-tables in Table 5. Owing to space limitations, we do not show the learning curves. Rather, we show results for three selected points on each learning curve, which correspond to the three major columns in each sub-table. For instance, for Abortion, the three selected points correspond to training set sizes of 300, 600, and 1000. Within each major column there are six columns corresponding to the six learning models, among which the two fine-grained models are marked with the

Configuration	300						600						1000					
	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF
W	57.1	57.6*	59.1‡	59.2*	60.1‡	60.2	59.2	60.1*	60.0	60.9*	61.9‡	62.7	61.1	61.7*	62.9‡	63.1*	64.3‡	65.3
A	57.5‡	57.9	59.6‡‡	59.7*	60.5‡‡	60.4	59.5‡	60.3*	60.0	61.0*	61.9‡	62.9‡*	61.3‡	61.8‡*	63.1‡	63.2*	64.4‡	65.9‡*
A+FS	59.8‡	59.9‡	61.7‡‡	61.8‡*	63.6‡‡	61.5	62.1‡	61.9‡	62.1‡	63.4‡*	64.4‡‡	65.1‡	63.1‡	62.7	64.2‡‡	64.7‡*	65.3	64.9
A+FS+N	62.6‡	61.8‡	63.4‡‡	64.2‡*	65.0‡‡	63.4‡*	63.9‡	63.5‡	63.9‡	65.5‡*	66.6‡‡	66.0‡	64.6‡	64.3‡	65.2‡‡	66.7‡*	67.5‡‡	67.5‡
A+FS+N+AC	70.0‡	69.7‡	70.3‡	71.7‡*	72.5‡‡	70.6‡*	71.0‡	70.9‡	71.4‡	71.9‡*	73.6‡‡	71.5‡*	73.5‡	73.3‡	74.1‡‡	74.0‡*	75.1‡‡	74.7‡

(a) Abortion

Configuration	300						600						1000					
	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF
W	58.3	58.3	59.5‡	60.0*	61.3‡	63.4*	61.1	60.7	62.4‡	62.7*	63.6	65.1*	62.5	62.1	63.6‡	63.2*	64.5‡	65.6*
A	59.0‡	59.2‡	59.7‡	60.2*	61.7‡‡	63.2*	61.8‡	61.4‡	62.6‡	62.4*	63.7‡	65.3*	62.6	62.4‡	63.5‡	63.8‡*	64.9‡‡	65.8*
A+FS	60.8‡	60.6‡	61.6‡‡	62.4‡*	63.5‡‡	64.8‡*	63.1‡	62.8‡	64.2‡‡	64.1‡*	64.9‡‡	66.2‡*	64.0‡	64.1‡	64.8‡	65.0‡*	66.3‡‡	66.8‡
A+FS+N	63.2‡	63.2‡	64.8‡‡	64.7‡*	66.0‡‡	65.9‡	64.5‡	64.8‡	65.8‡‡	66.2‡*	67.5‡‡	66.7	64.9‡	65.2‡	65.9‡	66.8‡*	68.2‡‡	67.6‡
A+FS+N+AC	65.4‡	65.3‡	66.7‡‡	66.5‡*	68.6‡‡	67.5‡*	66.0‡	66.2‡	67.2‡‡	67.8‡*	69.5‡‡	68.5‡*	66.9‡	67.0‡	67.9‡‡	68.9‡*	71.1‡‡	69.9‡*

(b) Gay Rights

Configuration	200						400						700					
	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF
W	56.2	56.3	58.3‡	58.1*	60.2‡	58.6*	57.3	57.7	59.2‡	59.5*	61.4‡	61.2	57.9	58.1	60.3‡	60.2*	62.0‡	62.9
A	56.6‡	56.7‡	58.1‡	58.0*	60.1‡	59.0‡*	57.4	57.8	59.5‡	59.7*	61.7‡	61.2	58.1	58.2	60.6‡	60.1*	62.2‡	63.2‡*
A+FS	58.7‡	58.9‡	60.6‡‡	60.2‡*	62.4‡‡	61.1‡*	59.3‡	59.7‡	61.9‡‡	61.8‡*	63.6‡‡	63.2‡	60.0‡	60.2‡	62.7‡‡	62.1‡*	64.3‡‡	64.2‡
A+FS+N	61.7‡	62.0‡	63.9‡‡	63.6‡*	65.7‡‡	64.6‡*	62.5‡	62.5‡	65.1‡‡	64.9‡*	67.1‡‡	66.1‡*	63.4‡	63.5‡	65.8‡‡	65.5‡*	68.0‡‡	67.1‡*
A+FS+N+AC	64.6‡	64.7‡	67.3‡‡	67.3‡*	69.8‡‡	68.7‡*	65.6‡	65.5‡	68.6‡‡	69.2‡*	70.7‡‡	70.3‡	66.6‡	67.0‡*	69.1‡‡	70.0‡*	71.9‡‡	71.1‡*

(c) Obama

Configuration	100						300						500					
	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF	SVM	NB	NB <sup>F</sup>	HMM	HMM <sup>F</sup>	CRF
W	63.5	63.9	64.7	65.5*	67.0‡	66.4	64.3	64.5	65.8‡	67.0*	68.3‡	68.7	66.0	65.9	67.1‡	68.5*	69.8‡	70.5
A	64.1‡	64.2	65.1‡‡	66.1‡*	67.2‡	66.7	65.5‡	65.6‡	66.4‡	67.3*	68.6‡	69.0	66.9‡	66.8‡	67.3	69.0‡*	70.1‡	70.8
A+FS	66.2‡	66.4‡	67.2‡‡	68.3‡*	69.1‡‡	68.5‡	67.7‡	67.9‡	68.6‡	70.0‡*	71.0‡‡	71.1‡	69.0‡	69.3‡	69.2‡	71.6‡*	72.0‡	72.6‡
A+FS+N	68.4‡	68.6‡	69.8‡‡	70.5‡*	71.8‡‡	70.6‡*	69.9‡	70.1‡	71.0‡‡	72.5‡*	73.3‡‡	73.1‡	71.3‡	71.1‡	72.0‡‡	73.7‡*	74.6‡‡	74.7‡
A+FS+N+AC	69.3‡	69.5‡	70.9‡‡	71.4‡*	72.7‡‡	71.6‡*	71.0‡	71.1‡	71.9	73.7‡*	74.2‡	74.2‡	72.2‡	72.4‡	73.4‡‡	74.9‡*	75.7‡‡	75.4‡

(d) Marijuana

Table 5: Five-fold cross-validation accuracies for the four domains.

superscript ‘F’. There are five rows in each sub-table. The ‘W’ row shows the results when only n-gram features are used. The ‘A’ row shows the results when Anand et al.’s (2011) features are used. The ‘A+FS’ row shows the results when both Anand et al.’s features and frame-semantic features are used. The last two rows show the results when noisily labeled documents and author constraints are added incrementally to A+FS.

To determine statistical significance, we conduct paired  $t$ -tests ( $p < 0.05$ ). These significance tests can be divided into three groups. The first group aims to determine whether the performance difference between the two systems shown in consecutive rows in a given column is statistically significant. If a number is marked with a dagger (†), it means that the performance difference between the corresponding system and the one in the previous row is statistically significant. The second group aims to determine whether the performance difference between two learning models are significant. We tested significance for three pairs of learning models: (1) SVM and NB; (2) NB and HMM; and (3) HMM<sup>F</sup> and CRF. If a number is marked with an asterisk (\*), it means

that the performance difference between the corresponding learning model and the one in the same pair is statistically significant.<sup>10</sup> The third group aims to determine whether the performance difference between NB/HMM and the corresponding fine-grained version of the model is statistically significant. If a number for a fine-grained model (NB<sup>F</sup>, HMM<sup>F</sup>) is marked with a double dagger (‡), it means that the performance difference between the model and its corresponding coarse-grained version (NB, HMM) is significant.

### 4.3 Discussion

**Q:** Can we improve performance by increasing the number of stance-labeled posts in the training set?

**A:** Yes. Keeping other factors constant, as we increase the number of (cleanly labeled) training posts from 100 to 500, we see significant improvements on all four domains: accuracies increase by 1.5, 2.4, 2.0, and 3.1 points for ABO, GAY, OBA, and MAR, respectively. As we further increase the number of training posts from 500 to 1000, we see

<sup>10</sup>If a number under the NB column is marked with an asterisk, it means that the performance difference between NB and SVM is significant.

another significant rise in performance: accuracies improve by 2.7 and 1.3 points for ABO and GAY, respectively. For ABO, GAY, and OBA, increasing the training set size seems to have a more positive impact on systems employing a simple feature set (W) than on those employing richer feature sets. Other than that, the degree of improvement does not seem to be dependent on the complexity of the model and the richness of the feature set.

**Q:** Which model is better, NB or SVM?

**A:** There is no clear winner. Other factors being equal, SVM beats NB significantly in 17% of the cases, NB beats SVM significantly in 27% of the cases, and the two are statistically indistinguishable in the remaining cases. Neither generative models nor discriminative models seem to have an advantage over the other for this task.

**Q:** Are the sequence models better than their non-sequence counterparts?

**A:** Yes. Comparing NB and HMM, we see that HMM consistently outperforms NB significantly, with improvements ranging from 1.6 to 2.2 points for the four domains. Now, comparing HMM and CRF, we see that while CRF does not always perform significantly better than HMM, in no case does it perform significantly worse.<sup>11</sup> Taken together, both sequence learners perform significantly better than NB. Since NB and SVM perform at the same level, we can conclude that sequence models indeed offer better performance.

**Q:** Are the fine-grained models better than their coarse-grained counterparts?

**A:** Considering HMM and HMM<sup>F</sup>, the answer is yes: HMM<sup>F</sup> beats HMM significantly by 1.1 to 2.1 points for the four domains. Considering NB and NB<sup>F</sup>, the answer is mostly yes: NB<sup>F</sup> beats NB significantly by 1.2 to 2.3 points for GAY and OBA respectively. For the remaining domains, NB<sup>F</sup> performs significantly better than NB in most cases, especially when the n-gram feature set and the Anand et al.'s feature set are used.

**Q:** Which is the best model?

**A:** HMM<sup>F</sup> and CRF achieve the best results, but there is no clear winner between them. Other factors being equal, CRF beats HMM<sup>F</sup> significantly in 26% of the cases, HMM<sup>F</sup> beats CRF significantly in 21% of the cases, and the two are statistically indistinguishable in the remaining cases.

<sup>11</sup>Significance test results between HMM and CRF are not shown in Table 5 due to space limitations.

**Q:** Is Anand et al.'s feature set (A) stronger than the n-gram feature set (W)?

**A:** Although the A systems generally yield small improvements (<1%) over the corresponding W systems, only 42% of those cases represent significant improvements. On the other hand, the W systems beat the corresponding A systems less than 15% of the times, and less than 10% of those cases represent significant improvements.

**Q:** Are frame-semantic features (FS) useful?

**A:** Yes. Apart from a few cases in ABO, the A+FS systems significantly outperform the corresponding A systems by 1.5–2.2 accuracy points for the four domains.

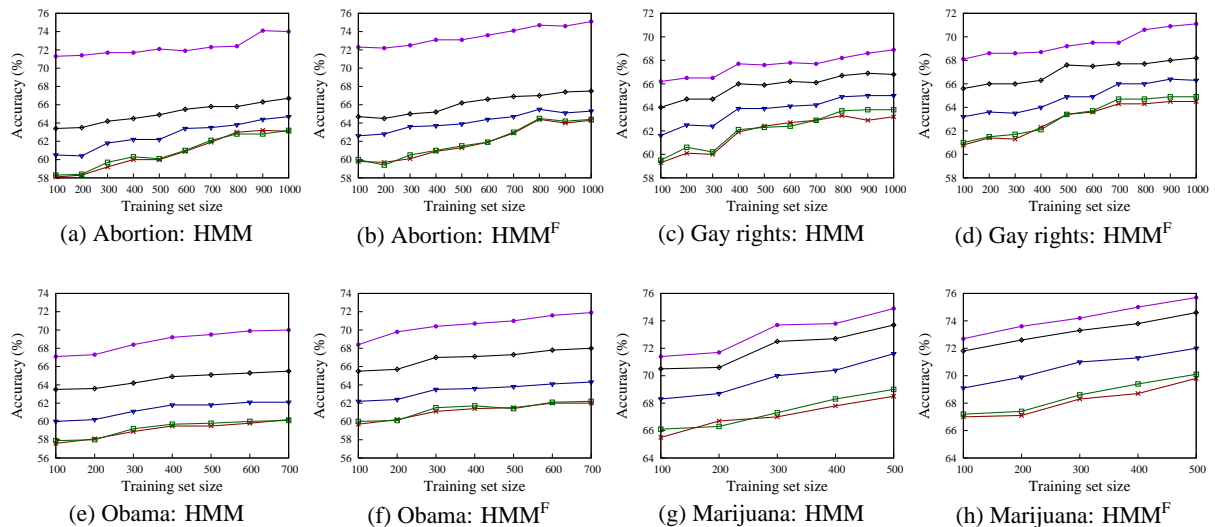
**Q:** Does using noisily labeled documents help improve performance?

**A:** Yes. Comparing A+FS and A+FS+N, we see that employing noisily labeled documents consistently yields a significant improvement of 1.8 to 3.3 points for the four domains, regardless of which learning model is used. For ABO and GAY, the improvement that we obtain out of the noisy data decreases as we increase the number of (cleanly labeled) debate posts. However, for OBA and MAR, we do not see such diminishing returns. This could be explained by the difference in the quality of the noisily labeled documents acquired for the different domains, but additional experiments are needed to determine the reason.

**Q:** Do ACs have different degrees of impact in different domains? If so, why?

**A:** Yes, ACs do seem to have different degrees of impact in different domains: on average, the addition of ACs yields a 7% improvement in ABO, a 2-3% improvement in GAY, a 4% improvement in OBA, and a <1% improvement on MAR. We hypothesize that this difference has to do with the percentage of test posts to which ACs can be applied successfully (i.e., an incorrect stance prediction will be turned into a correct one after applying ACs). To test this hypothesis, we take a closer look at two runs, an ABO run where HMM<sup>F</sup> is trained on 1000 posts and a MAR run where HMM<sup>F</sup> is trained on 500 posts. If our hypothesis is correct, then a larger fraction of the test posts in ABO should become correctly classified after the application of ACs. Indeed, the results are consistent with our hypothesis: we find that more than 8% of the test posts in ABO become correctly classified after applying ACs, while the corresponding number for MAR is less than 2%.





## Appendix: Learning Curves

The eight graphs above are the learning curves for HMM and HMM<sup>F</sup> for the four domains. The five curves in each graph correspond to the configurations in the five rows of each sub-table in Table 5. In each graph, the best-performing configuration is A+FS+N+AC, which is followed by A+FS+N and then A+FS. There is no clear winner between W and A, but the latter tends to outperform the former as the amount of training data increases.

## References

- R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. 2003. Mining newsgroups using networks arising from social behavior. *WWW*.
- P. Anand, M. Walker, R. Abbott, J. E. Fox Tree, R. Bowmani, and M. Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. *WASSA*.
- C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet project. *COLING/ACL*.
- A. Balahur, Z. Kozareva, and A. Montoyo. 2009. Determining the polarity and source of opinions expressed in political debates. *CICLing*.
- M. Bansal, C. Cardie, and L. Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. *COLING 2008: Posters*.
- O. Biran and O. Rambow. 2011. Identifying justifications in written dialogs. *ICSC*.
- C. Burfoot, S. Bird, and T. Baldwin. 2011. Collective classification of congressional floor-debate transcripts. *ACL-HLT*.
- D. Das, N. Schneider, D. Chen, and N. A. Smith. 2010. SEMAFOR 1.0: A probabilistic frame-semantic parser. Carnegie Mellon University Technical Report CMU-LTI-10-001.
- 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)*, 39:1–38.
- K. S. Hasan and V. Ng. 2012. Predicting stance in ideological debate with rich linguistic knowledge. *COLING 2012: Posters*.
- T. Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.
- J. D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*.
- A. K. McCallum. 2002. Mallet: A machine learning for language toolkit.
- A. Murakami and R. Raymond. 2010. Support or oppose? Classifying positions in online debates from reply activities and opinion expressions. *COLING 2010: Posters*.
- A. Y. Ng and M. I. Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *NIPS*.
- T.-V. T. Nguyen and A. Moschitti. 2011. Joint distant and direct supervision for relation extraction. *IJC-NLP*.
- S. Somasundaran and J. Wiebe. 2010. Recognizing stances in ideological on-line debates. *CAAGET*.
- M. Thomas, B. Pang, and L. Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. *EMNLP*.
- M. Walker, P. Anand, R. Abbott, and R. Grant. 2012. Stance classification using dialogic properties of persuasion. *NAACL-HLT*.
- Y.-C. Wang and C. P. Rosé. 2010. Making conversational structure explicit: Identification of initiation-response pairs within online discussions. *NAACL-HLT*.
- A. Yessenalina, Y. Yue, and C. Cardie. 2010. Multi-level structured models for document-level sentiment classification. *EMNLP*.

# University Entrance Examinations as a Benchmark Resource for NLP-based Problem Solving

**Yusuke Miyao**

National Institute of Informatics  
Graduate University for Advanced Studies  
yusuke@nii.ac.jp

**Ai Kawazoe**

National Institute of Informatics  
zoeai@nii.ac.jp

## Abstract

This paper describes a corpus comprised of university entrance examinations, which is aimed to promote research on NLP-based problem solving. Since entrance examinations are created for quantifying human ability of problem solving, they are a desirable resource for benchmarking NLP-based problem solving systems. However, as entrance examinations involve a variety of subjects and types of questions, in order to pursue focused research on specific NLP technologies, it is necessary to break down entire examinations into individual NLP subtasks. For this purpose, we provide annotations of question classifications in terms of answer types and knowledge types. In this paper, we also describe research issues by referring to results of question classification, and introduce two international shared tasks that employed our resource for developing their evaluation data sets.

## 1 Introduction

This paper introduces natural language corpora whose source texts are taken from university entrance examinations. This resource has been developed aiming at benchmarking NLP systems for problem solving. In general, entrance examinations are mostly described in natural language, and the goal of reading the text is clear, viz., to solve questions. Therefore, this is an ideal resource for evaluating end-to-end NLP systems that read natural language text, perform some information processing, and output answers.

University entrance examinations have several desirable features to be used for benchmarking NLP-based problem solving. They are carefully designed for empirically quantifying a certain

ability of high-school-level students. Therefore, it is not a trivial task for NLP systems to solve university entrance examinations. On the other hand, it is guaranteed that required knowledge is fairly restricted, and legitimate solutions always exist. Despite such artificial restrictions, investigating the entire process of solving entrance examinations is meaningful, because it is expected to reveal true contributions of current NLP technologies to human-like problem solving tasks. In addition, evaluation results are intuitively understandable, and can be compared directly with human performance. This provides us with empirical evidence for analyzing the relationships between human intelligence and artificial intelligence.

While it is now clear that university entrance examinations are a useful resource for NLP benchmarking, it is also true that they will not be appropriate for focusing on individual NLP tasks, because they involve a variety of subjects and types of questions. It is almost hopeless to invent a single clever algorithm that can solve all types of questions. Therefore, it is necessary to break down entire examinations into NLP subtasks that can be investigated solely. For this aim, we annotate classifications of questions, which allow us to isolate specific NLP subtasks for focused research. An important point here is that, question classification allows us to extract individual NLP tasks, but, at the same time, their contributions to entire performance are always accessible. Therefore, our resource is inherently different from NLP resources that focus on monolithic NLP tasks/applications in nature, such as parallel corpora for machine translation research and evaluation data sets for question answering. Owing to question classifications, subsets of our resources have been adopted in international shared tasks for recognizing textual entailment and reading comprehension, which will also be mentioned in this paper.

Standardized tests for high-school-level stu-

dents are widely accepted in the world; examples include SAT (U.S.), Baccalauréat (France), Suneung (Korea), Gao Kao (China), and Center Test (Japan). In this work, we collect source texts from examinations of Center Test in Japan. Center Test has additional advantages as a NLP resource, because texts are free from copyright issues, and questions are given in a multiple-choice style, which allows for automatic evaluation.

The contributions of this paper are summarized as below:

- Describes details of the design of resources developed from university entrance examinations.
- Classifies questions from the NLP point of view, and discusses research issues involved.
- Introduces present use cases of our resources to show their effectiveness.

The resources introduced in this paper are made available for research purposes. As we will see below, this resource involves a variety of research issues in NLP and related AI technologies, and thus collaborative research based on such open resources is indispensable.

## 2 Motivation

Current NLP corpora can be classified into two types. One is to focus on specific fundamental NLP technologies, such as Penn Treebank (Marcus et al., 1993) developed for parsing research. The other is application-oriented data sets, meaning that corpora are used for evaluating specific NLP applications, such as machine translation and question answering (Voorhees and Buckland, 2012; Kando et al., 2011). However, despite significant advancement achieved by these resources, it is still unclear how far current NLP technologies have approached human intelligence, in particular, about the ability of generic problem solving. In the current NLP, research topics are inherently determined when corpora are developed, and there is no room for investigating performances of NLP technologies from a holistic view.

Our primary motivation to develop a corpus of university entrance examinations is to provide an open data set that encourages research on end-to-end NLP systems for problem solving. By investigating the entire process of solving various types

of examinations, we expect to recognize contributions of current NLP technologies and methodologies for integrating them from a holistic view.

For this purpose, university entrance examinations have several advantages as a benchmark, as explained below.

**Open but restricted real-world task** Since university entrance examinations are developed for empirically quantifying a certain ability of humans, solving them is not a toy task. However, because questions must be solvable by high-school students, this task requires much smaller knowledge space than contemporary NLP applications such as Web-scale question answering. Therefore, we can focus on algorithms of problem solving rather than relying on huge data.

**Fair and clear evaluation criteria** Intrinsically, standardized tests of university entrance examinations are carefully designed to guarantee fairness. To be more concrete, it is guaranteed that correct answers always exist, and everybody agrees with correct answers. This means that gold standard data is given at almost perfect agreement, which is an ideal feature as a benchmark. In addition, questions of Center Test are given in a multiple-choice style, which allows for automatic evaluation.

**Necessity of heterogeneous NLP tasks** Since university entrance examinations are aimed at quantifying various aspects of human intelligence, various forms of questions are developed in a variety of subjects. Therefore, to develop an end-to-end system to solve questions, multiple NLP components have to work in a collaborative manner. In some cases, they have to be connected to non-NLP components, such as mathematical solvers and ontology-based inference engines. Thus, investigating entrance examinations promotes interdisciplinary research within NLP, as well as with outside of NLP.

This also indicates the difficulty of focused research on individual NLP technologies. Therefore, we provide annotations for question classification, which enables us to extract a subset of examinations that is relevant to focused NLP tasks (see Section 3 and 4).

**Comparison to human performance** In our framework, overall performance of NLP systems is quantified as *scores*, which are directly comparable with human performance. We can therefore

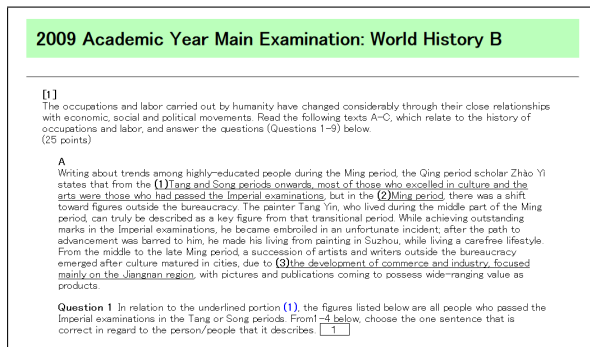


Figure 1: A screenshot of university entrance examination (2009 Center Test World History B)

recognize advantages and disadvantages of current NLP technologies compared to human problem solving. We can also empirically investigate relationships between NLP technologies and human ability of language understanding. Although NLP systems do not necessarily imitate human language processing, it is scientifically interesting to explore such relationships.

A possible criticism to our resource would be on its practical value. It is obvious that solving entrance examinations by NLP systems does not have any practical value. However, our intention is on the investigation of the whole process of problem solving. We believe that such holistic analysis of problem solving contributes to better understanding of current NLP technologies.

Another criticism would be on the reason why we focus on *university* entrance examinations, rather than easier questions, such as elementary school tests and TOEFL-like English tests. In this respect, we argue that university entrance examinations are most appropriate as a NLP benchmark. In preliminary investigation, we found that easy tests like elementary school tests rely more on generic knowledge and common sense, probably because the knowledge space and the vocabulary that can be used are too restricted. On the other hand, examinations in more expertized areas, such as medical license tests, are more uniformly developed and involve a less variety of NLP subtasks.

### 3 Resources

This section describes details of the resources we have developed. As we stated in Section 2, our primary aim is to investigate the entire process of problem solving related to natural language understanding. Therefore, we prepare data sets in which

```
<exam subject="World History B (main)" year="2009">
  <title>
    2009 Academic Year Main Examination: World History B
  </title>
  <question id="Q1" minimal="no">
    <label>[1]</label>
    <instruction>
      The occupations and labor carried out by humanity have
      changed considerably through their close relationships
      with economic, social and political movements...
    </instruction>
    <data id="D0" type="text">
      <label>A</label>
      Writing about trends among highly-educated people during
      the Ming period, the Qing period scholar Zhao Yi states
      that from the (1)Tang and Song periods onwards, most of
      those who excelled in culture and the arts were those
      who had passed the imperial examinations, but in the
      (2)Ming period, there was a shift toward figures outside
      the bureaucracy. The painter Tang Yin, who lived during
      the middle part of the Ming period, can truly be
      described as a key figure from that transitional period.
      While achieving outstanding marks in the imperial
      examinations, he became embroiled in an unfortunate
      incident, after the path to advancement was barred to
      him, he made his living from painting in Suzhou, while
      living a carefree lifestyle. From the middle to the late
      Ming period, a succession of artists and writers outside
      the bureaucracy emerged after culture matured in cities,
      due to (3)the development of commerce and industry,
      focused mainly on the Jiangnan region, with pictures and
      publications coming to possess wide-ranging value as
      products.
    </data>
    <question id="Q2" minimal="yes" answer_type="sentence"
      knowledge_type="KS">
      <label>Question 1</label>
      <instruction>
        In relation to the underlined portion <ref target="U1">
          (1)</ref>, the figures listed below are all people
          who passed the Imperial examinations in the Tang or
          Song periods. From 1-4 below, choose the one
          sentence that is correct in regard to the person/people
          that it describes.
      </instruction>
      <ansColumn id="A1">1</ansColumn>
      <choices anscol="A1">
        <choice><cNum>(1)</cNum>Ouyang Xiu and Su Shi are
          writers representative of the Tang period.</choice>
        <choice><cNum>(2)</cNum>Yan Zhenqing is a calligrapher
          representative of the Song period.</choice>
        <choice ra="yes"><cNum>(3)</cNum>Wang Anshi, who
          lived during the Song period, carried out reforms
          called the New Policies (xin fa).</choice>
        <choice><cNum>(4)</cNum>Qin Hui came into conflict
          with the party in favor of war, concerning the
          relationship with the Yuan.</choice>
      </choices>
    </question>
    ...
  </question>
```

Figure 2: XML data of university entrance examination (2009 Center Test World History B)

	Exam (orig.)	Exam (Eng.)	Textbook
# subjects	11	5	6
# files	571	25	11
# questions	17260	771	N/A
# sentences	79211	5236	35183

Table 1: Statistics of corpora

non-linguistic structures are already solved. Figure 1 shows a screenshot of an actual examination, while Figure 2 shows its XML version, where document structures, such as *instruction* and *question* are already given. Relying on these structure annotations, we can easily extract relevant texts for research, such as questions of interest, and their related instructions, etc.

Basically, all the questions of Center Test are given in a multiple-choice style. The answer data is also given in the XML format. Given answer data, it is almost trivial to compute examination scores, while we also provide tools for automatic evaluation and visualization.

#### 3.1 Corpora

In this work, we collected PDFs and source texts from National Center Test for University Admis-

	Exam (orig.)	Exam (Eng.)	Textbook
Document structure	✓	✓	✓
Question types	✓	✓	
Technical terms	✓		✓
Dependency trees			✓
Coreferences			✓

Table 2: Summary of annotated resources

sion in Japan (a.k.a. Center Test).<sup>1</sup> Center Test is a nation-wide standardized test for university admission in Japan, and almost all high-school students who aim to enter a university in Japan take this exam. Therefore, questions are carefully designed in order to accurately quantify achievement levels of high-school students.

Table 1 shows a summary of source texts. The Center Test corpus includes examination texts from eleven subjects, namely, *World History*, *Japanese History*, *Modern Society*, *Politics & Economics*, *Ethics*, *Physics*, *Chemistry*, *Biology*, *Mathematics*, *Japanese (native language)*, and *English (foreign language)*, used in the years from 1990 to 2011.<sup>2</sup> In each year, a single main examination and a couple of additional examinations are available. In total, we have obtained 571 examinations, each of which contains 30-50 questions. The numbers of questions and sentences are also shown in the table, indicating a comprehensive amount as a corpus for NLP research.

While original texts are in Japanese (except for English tests), a part of examinations of World History, Politics & Economics, Physics, Chemistry, and Biology, are translated into English, in order to allow researchers to work on English NLP as well as cross-lingual NLP.

In addition to examinations, we collected textbooks of World History, Japanese History, Modern Society, Politics & Economics, Ethics, and Biology. Questions in these subjects often ask to recognize *facts*, such as historical facts and biological processes. Because textbooks describe such facts, we can use textbooks as knowledge sources for solving such questions. In fact, these textbooks were adopted as knowledge sources in a shared task on recognizing textual entailment (Section 5).

For these text data, we annotated document structures, question types, technical terms, dependency trees, and coreferences (see Table 2), which are explained in the consecutive sections.

<sup>1</sup><http://www.dnc.ac.jp/>

<sup>2</sup>Examination corpora of Geography, Geology, and General Science are under construction.

question	A question region including outermost question areas and minimal areas. An ID is assigned to each element. Question regions that do not include other question regions are given the attribute <code>minimal="yes"</code> , indicating smallest units of questions.
instruction	A statement or an instruction for a question.
data	Data provided to test-takers of reference, including not only texts but also images, tables, graphs, etc.
label	A label such as section numbers, question numbers, identifiers of text fragments, etc.
ansColumn	An identifier of an answer column. Each answer column is given a unique ID, which is referred to in answer data.
choices	A set of choices.
choice	An individual choice. The attribute <code>ra="yes"</code> denotes correct choices.
cNum	An identifier of a choice.
ref	A symbol that refers to another text fragment, such as underlined texts. A referred text fragment is denoted by the attribute <code>target</code> .
uText	An underlined text fragment. A unique ID is assigned when the text fragment is referred to by <code>ref</code> .

Table 3: Document structure tags

### 3.2 Document Structure Annotation

Examination texts are highly structured, while the automatic recognition of document structures is still a challenging task (Schäfer and Weitz, 2012). Therefore, our resource is provided with human-annotated document structures in the form of XML, as shown in Figure 2. Table 3 shows an excerpt of XML tags used for the annotation.<sup>3</sup> In addition to document structures, texts are also annotated with extra-linguistic markups, such as underlines (`uText`) and references (`ref`).

Owing to the document structure annotations, users can easily extract questions and relevant text regions. For example, a complete list of individual questions can be obtained by extracting elements `<question minimal="yes">`, and their corresponding answer columns and choices can also be extracted easily. Furthermore, text fragments referred to by a label like “(1)” can be obtained by following the attribute `target` of `ref` (see the example in Figure 2).

Formulas play crucial roles in examinations of Science and Mathematics. Although understanding of semantics of formulas is indispensable, the

<sup>3</sup>The complete list and the definitions of tags are provided together with annotated corpora.

<i>Answer types</i>	
sentence	Choices are described by sentences.
term	Choices are described by terms (e.g. person names).
image	Choices are represented by images or parts of an image.
formula	Choices are represented by formulas.
combination	Choices are described by a combination of sentences, terms, etc.
<i>Knowledge types</i>	
KS	An external knowledge source (e.g. textbooks) is required.
RT	Reading comprehension of a text given within a question is required.
IC	Image comprehension is necessary.
GK	General knowledge is required.
DM	Domain-specific inference (e.g. laws of dynamics) is required.

Table 4: Top-level categories for question classification

From 1-4 below, choose the one sentence that is correct in regard to the person/people that it describes.  
(1) Ouyang Xiu and Su Shi are writers representative of the Tang period.  
(2) Yan Zhenqing is a calligrapher representative of the Song period.  
(3) Wang Anshi, who lived during the Song period, carried out reforms called the New Policies (xin fa).  
(4) Qin Hui came into conflict with the party in favor of war, concerning the relationship with the Yuan.

Figure 3: A true-or-false question

semantic analysis of formulas is not trivial and is beyond the scope of NLP research. Therefore, we marked up all formulas that appear in examination texts with MathML.

### 3.3 Question Type Annotation

As mentioned in Section 2, university entrance examinations involve a variety of NLP subtasks, which prevents us from focusing on individual NLP tasks. In order to extract questions of interest, we annotate each question with classification categories. By extracting questions assigned specific categories, we can obtain a subset of examinations on which isolated NLP tasks can be studied.

Table 4 shows a subset of top-level categories for question classification. Questions are classified according to two perspectives. The *answer type* specifies the format of answers. For example, if choices are presented with a sentence, it is assigned the category *sentence*, which typically indicates *true-or-false questions* as exemplified in Figure 3. If *term* is assigned, the question is likely to be a *factoid-style question*. These categories are further classified into sub categories; for example,

*term* is divided by term categories (e.g. *person names*), while *combination* is further classified with elements of combinations. In total, 25 answer type categories are annotated.

The *knowledge type* describes the types of knowledge that are necessary to solve the question. While Table 4 shows representative top-level categories, they are further divided into fine-grained categories, and, in total, 90 knowledge type categories are annotated. For example, *KS* indicates that to answer the question requires referring to an external knowledge source like textbooks (e.g. Figure 3). This type of questions typically appear in examinations of Social Studies. *RT* indicates a similar type of questions, but necessary information is given as a text within an examination. Therefore, reading comprehension is necessary. *DM* means domain-specific inference is necessary depending on a subject. Individual domains are annotated with finer-grained categories, like *physical mechanics* and *electromagnetics*. For example, to solve questions of physical dynamics, calculation of formulas based on laws of dynamics is required. *GK* indicates any other type of knowledge, such as *typical situations and reactions in a restaurant*. Since the knowledge space is not strictly restricted, we suppose this is the most difficult type of questions for NLP systems.

In Section 4, we will discuss research issues involved in our resource, by observing results of question classification described here.

### 3.4 Linguistic Annotation

In addition to document structures and question classifications, we have developed resources annotated with technical terms, dependency trees, and coreference relations, in order to support research on fundamental NLP tools.

Technical terms are annotated to examinations and textbooks of World History, Japanese History, Modern Society, Politics & Economics, Ethics, and Biology. These subjects are selected because, as we will see in Section 4, a majority of questions in these subjects are either true-or-false or factoid-style questions, which can be approached by searching textbooks for relevant evidences. In such a scenario, technical terms are crucial keys for accurate search. For example, to solve the question shown in Figure 3, it is necessary to correctly recognize relationships among named entities like *Ouyang Xiu* and *the Tang period*.

	W. Hist.	J. Hist.	M. S.	P. & E.	Ethics	Bio.
instance	8864 (52.1)	5876 (35.5)	2558 (24.3)	2279 (22.6)	2556 (28.8)	90 (0.6)
class	5592 (32.9)	7808 (47.2)	5084 (48.4)	4779 (47.3)	1237 (14.0)	2382 (15.6)
both	2557 (15.0)	2848 (17.2)	2867 (27.3)	3039 (30.1)	5072 (57.2)	12790 (83.8)
# terms	17013 (100.0)	16532 (100.0)	10509 (100.0)	10097 (100.0)	8865 (100.0)	15262 (100.0)
# sentences	5797	5571	3674	3352	3245	4215

Table 5: Statistics of technical term annotations

	W. Hist.	J. Hist.	M. S.	P. & E.	Ethics
True-or-false question	1854 (73.6)	1308 (55.6)	1102 (79.5)	805 (88.6)	656 (81.8)
Factoid question	464 (18.4)	557 (23.7)	192 (13.9)	62 (6.8)	128 (16.0)
Reading comprehension	102 (4.0)	146 (6.2)	43 (3.1)	3 (0.3)	88 (11.0)
General knowledge	1 (0.0)	0 (0.0)	92 (6.6)	8 (0.9)	114 (14.2)
Image comprehension	222 (8.8)	198 (8.4)	111 (8.0)	101 (11.1)	17 (2.1)
# questions	2519 (100.0)	2351 (100.0)	1386 (100.0)	909 (100.0)	802 (100.0)

Table 6: Classification of questions (Social Studies)

We analyzed our corpus of examinations and textbooks, and developed an ontology of technical terms, which involves 72 categories. Their occurrences in examinations and textbooks are annotated manually. Table 5 shows statistics of technical term annotations on examinations.<sup>4</sup> Annotated terms not only include typical named entities (i.e. *instances*) like *person names*, but also include *class concepts* that describe domain-specific abstract terms, such as *genetic trait*. Several categories may include both instance terms and class terms (denoted as “both” in the table); for example, the category *artwork* includes *Isenheim Altarpiece* (instance) and *miniature* (class). Interestingly, the distributions of terms imply characteristics of each subject; for example, World History concerns named entities, while Biology is more focused on abstract concepts.

Dependency trees and coreferences are annotated in order to assess the performance of fundamental NLP tools including dependency parsers and coreference resolution systems. It is expected that these NLP tools work reasonably well on texts of examinations and textbooks, because in general texts in these domains are written in an unambiguous and easy-to-understand way. Currently, we have annotated a subset of a textbook of World History, and will extend the data as necessary.

#### 4 Analysis of Questions

This section discusses research issues involved in solving the university entrance examinations, by analyzing question classification results. Table 6, Table 7, and Table 8 show the number of

questions and its ratio (shown in brackets) classified into each category, for examinations of Social Studies, Science, and English/Japanese, respectively.<sup>5</sup> These classifications are obtained from answer type and knowledge type annotations introduced in Section 3, while classification categories are summarized and reinterpreted for readability.

For Social Studies (Table 6), it is obvious that most of the questions can be classified into *true-or-false* and *factoid-style questions*. Low ratios of *reading comprehension* and *general knowledge* indicate that most of the questions can be solved only by referring to external knowledge sources. This is promising, because current question answering methods and/or search-based methods would suffice. As we will see in Section 5, these types of questions have already been tackled in international shared tasks.

For Science subjects (Table 7), Biology looks similar to Social Studies, while results on Physics and Chemistry reveal different characteristics. Almost all questions in these subjects are annotated as *domain-specific inference*, indicating that simply referring to knowledge sources does not suffice, and inference engines, such as formula processing modules and ontology-based reasoning, will be required. In particular, nearly half of the questions in Physics are answered in *formulas*, indicating necessity of formula processing. The integration of NLP components with formula processing should be an interesting frontier.

Results on English and Japanese are totally different. They contain questions at different levels of difficulty. Questions that ask *lexi-*

<sup>4</sup>The statistics for textbooks is omitted for space limitation, but the tendency of the distribution is similar.

<sup>5</sup>The sum of ratios exceeds 100%, because a question might be classified into multiple categories.

	Physics	Chemistry	Biology
True-or-false question	390 (24.4)	578 (32.5)	938 (52.5)
Factoid question	239 (15.0)	367 (20.7)	564 (31.6)
Formula	683 (42.8)	399 (22.5)	136 (7.6)
Domain-specific inference	1594 (99.9)	1764 (99.3)	522 (29.2)
Reading comprehension	0 (0.0)	3 (0.2)	31 (1.7)
General knowledge	64 (4.0)	0 (0.0)	2 (0.1)
Image comprehension	1105 (69.3)	291 (16.4)	420 (23.8)
# questions	1595 (100.0)	1776 (100.0)	1767 (100.0)

Table 7: Classification of questions (Science)

	English	Japanese
Lexical knowledge	1085 (44.8)	778 (36.4)
Grammatical knowledge	703 (29.0)	126 (5.9)
Literature knowledge	0 (0.0)	36 (1.7)
Reading comprehension	892 (36.8)	872 (40.8)
Situation comprehension	1213 (50.1)	232 (10.8)
Rhetorical structure	0 (0.0)	173 (8.1)
Translation	0 (0.0)	465 (21.7)
Image comprehension	402 (16.6)	0 (0.0)
# questions	2423 (100.0)	2139 (100.0)

Table 8: Classification of questions (English and Japanese)

*cal/grammatical/literature knowledge* should be tractable for current NLP systems. However, a significant portion of questions involves *reading comprehension*, which is an outstanding problem in NLP. Research on reading comprehension is recently emerging (Peñas et al., 2011a; Peñas et al., 2011b), while the achievements are still far from satisfactory. Furthermore, English examinations involve a large portion of *situation comprehension* (e.g. selecting an appropriate conversation in a restaurant) and *image comprehension* (e.g. choosing an appropriate description of a given image), which are enormously difficult research issues. In this respect, achieving high scores in English tests can be an ultimate goal of the present effort.

While Mathematics is not shown in the tables, it is totally different from the subjects discussed above. Solving mathematics questions essentially consists of two components. One is natural language understanding, which converts text expressions into mathematical formulas, and the other is mathematical formula processing. Therefore, the primary research issues are to design an interface between the two components, and to increase the accuracy of the two components.

## 5 Use Cases

In addition to individual studies, two international shared tasks have adopted subsets of our resources for creating their evaluation data sets. Here we

<i>t</i> : In the period of Emperor Shenzong in the Baisong dynasty, Wang Anshi introduced and promulgated his reform policy (xin fa).
<i>h</i> : Wang Anshi, who lived during the Song period, carried out reforms called the New Policies (xin fa).

Figure 4: A text pair for recognizing textual entailment created from a World History examination.

briefly introduce these works, which prove the effectiveness of our resources for NLP research.

### 5.1 Recognizing Textual Entailment

The RITE task at the NTCIR conference is a shared task on recognizing textual entailment (Watanabe et al., 2013). RITE consisted of several subtasks, one of which adopted a subset of our resource as an evaluation data set. As described in Section 4, a significant portion of Social Studies consists of true-or-false questions, which can be solved by recognizing textual entailment relations. For example, Figure 3 shows a typical true-or-false question. Test-takers are required to find relevant facts from their knowledge, and judge whether each sentence is true or false. For NLP systems, this corresponds to finding an evidential text from a knowledge source like a textbook or Wikipedia, and judge whether a text fragment in the knowledge source *entails* each sentence. In fact, by extracting a relevant text from Wikipedia, we can create a text pair as shown in Figure 4, which can be used as evaluation data for recognizing textual entailment.

In the RITE task, true-of-false questions are extracted from four subjects, namely, World History, Japanese History, Modern Society, and Politics & Economies, while evidential texts are provided from Wikipedia and textbooks. In total, 510 text pairs are provided as a training set, and 448 pairs as a test set.

While the RITE task reveals that recognizing textual entailment can be applied directly to true-or-false questions, this is not the only solution



for this type of questions. Actually, Kanayama et al. (2012) demonstrated a method for applying a factoid-style question answering system to solve true-or-false questions, and evaluated their system using a World History portion of our resource. This reveals that a variety of approaches can be attempted to achieve the same goal, i.e., solving examinations.

## 5.2 Reading Comprehension

Shared tasks called Question Answering for Machine Reading Evaluation (QA4MRE) at the CLEF conferences (Peñas et al., 2011a; Peñas et al., 2011b) have been focusing on NLP technologies for reading comprehension tasks. In the task setting of QA4MRE, a short document is given, and systems are required to answer multiple-choice questions by reading the given document. Because given texts are small, methods that rely on huge texts as in typical question answering systems cannot be applied, while accurate and deep analysis of given texts is necessary. Original evaluation data sets for QA4MRE have been developed from scratch, focused on several topics like “Aids” and “Climate Change.”

In QA4MRE at CLEF 2013,<sup>6</sup> a pilot task that uses reading comprehension questions from English tests of our resource has been organized. The novelty of this pilot task is that questions are originally developed for assessing human English ability, rather than specifically developed for NLP system evaluation. Therefore, it is expected that various aspects of human natural language understanding appear in solving such questions.

## 6 Related Work

Recent advancement of empirical NLP owes much to language resources, such as annotated corpora and lexicons. Language resources to date have been developed specifically for focused NLP tasks, such as syntactic/semantic parsing, coreference resolution, and word sense disambiguation (Marcus et al., 1993; Kingsbury and Palmer, 2002; Hovy et al., 2006; Ide et al., 2010; Tateisi et al., 2005; Kawahara et al., 2002; Iida et al., 2007). Another type of corpora has been developed for evaluating NLP applications, such as machine translation and question answering, which are often provided in application-oriented evaluation campaigns (Voorhees and Buckland, 2012; Kando

<sup>6</sup><http://celct.fbk.eu/QA4MRE/>

et al., 2011; Catarci et al., 2012). In other words, the development of language resources is initiated by the demand for NLP tasks/applications. However, the resources presented in this paper are motivated in an opposite way. We start from texts that involve problem solving by humans, i.e., university entrance examinations, and by analyzing them we can identify NLP tasks that we have to tackle with. It can be said that the framework and the resources described in this paper provide another direction of NLP research.

NLP research that develops benchmark data from questions originally designed for evaluating human performance has also been emerging. For example, the Halo project (Angele et al., 2003) targeted Chemical tests, while IBM’s Deep QA (Ferrucci, 2012) employed factoid-style quizzes. However, their benchmark data sets are not open, and therefore collaborative research based on shared standard data cannot be pursued. Collaborative research is indispensable for our purpose, because entrance examinations involve a variety of NLP subtasks, and a single research group cannot solve the entire problem. Therefore, it is necessary to develop open resources as described in this paper.

## 7 Conclusion

We have introduced an NLP resource that is developed from university entrance examinations, aiming at the development and the evaluation of end-to-end NLP systems for problem solving. In total 571 examinations are collected from 11 subjects, involving 17260 individual questions, revealing a comprehensive resource for NLP benchmarking.

While the ultimate goal is to develop an integrated NLP system that can solve a wide range of questions, this also means it is difficult to focus on individual NLP subtasks. Therefore, we annotated question classifications so that users can extract fragments of the resource that are relevant to a focused NLP subtask. In fact, subsets of our resources have already been adopted by two international shared tasks, namely, NTCIR RITE for recognizing textual entailment, and CLEF 2013 QA4MRE, for reading comprehension.

In order to encourage collaborative and interdisciplinary research, the resources described in this paper are made available for research purposes.<sup>7</sup>

<sup>7</sup>The resource is available at <http://21robot.org/>.

## Acknowledgments

We gratefully acknowledge the National Center for University Entrance Examination, JC Educational Institute and Tokyo Shoseki Co., Ltd. for original Center Test data and textbook data.

## References

- J. Angele, E. Moench, H. Oppermann, S. Staab, and D. Wenke. 2003. Ontology-based query and answering in chemistry: OntoNova @ Project Halo. In *Proceedings of the Second International Semantic Web Conference*.
- T. Catarci, P. Forner, D. Hiemstra, A. Penas, and G. Santucci, editors. 2012. *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics*. Springer.
- D. A. Ferrucci. 2012. Introduction to “this is watson”. *IBM Journal of Research and Development*, 56(3.4).
- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of HLT/NAACL 2006*.
- 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.
- Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proceedings of Linguistic Annotation Workshop*, pages 132–139.
- Hiroshi Kanayama, Yusuke Miyao, and John M. Prager. 2012. Answering yes/no questions via question inversion. In *Proceedings of COLING 2012*.
- Noriko Kando, Daisuke Ishikawa, and Miho Sugimoto, editors. 2011. *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*.
- D. Kawahara, T. Kurohashi, and K. Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 8th Annual Meeting of the Association for Natural Language Processing*, pages 495–498. (In Japanese).
- P. Kingsbury and M. Palmer. 2002. From Treebank to PropBank. In *Proceedings of LREC 2002*.
- M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Anselmo Peñas, Eduard Hovy, Pamela Forner, Àlvaro Rodrigo, Richard Sutcliffe, Corina Forascu, and Caroline Sporleder. 2011a. Overview of QA4MRE at CLEF 2011: Question answering for machine reading evaluation. In *CLEF 2011 Labs and Workshop Notebook Papers*, pages 19–22.
- Anselmo Peñas, Eduard Hovy, Pamela Forner, Àlvaro Rodrigo, Richard Sutcliffe, Caroline Sporleder, Corina Forascu, Yassine Benajiba, , and Petya Osenova. 2011b. Overview of QA4MRE at CLEF 2012: Question answering for machine reading evaluation. In *CLEF 2011 Labs and Workshop Notebook Papers*.
- Ulrich Schäfer and Benjamin Weitz. 2012. Combining OCR outputs for logical document structure markup: technical background to the ACL 2012 contributed task. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pages 104–109.
- Y. Tateisi, A. Yakushiji, T. Ohta, and J. Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of IJCNLP 2005 Companion Volume*.
- E. M. Voorhees and Lori P. Buckland, editors. 2012. *Proceedings of the Twenty-First Text REtrieval Conference (TREC 2012)*.
- Yotaro Watanabe, Yusuke Miyao, Junta Mizuno, Tomohide Shibata, Hiroshi Kanayama, C.-W. Lee, C.-J. Lin, Shuming Shi, Teruko Mitamura, Noriko Kando, Hideki Shima, and Kohichi Takeda. 2013. Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. In *Proceedings of the 10th NTCIR Conference*.

# Linguistically Aware Coreference Evaluation Metrics

Chen Chen and Vincent Ng

Human Language Technology Research Institute

University of Texas at Dallas

Richardson, TX 75083-0688

{yzcchen, vince}@hlt.utdallas.edu

## Abstract

Virtually all the commonly-used evaluation metrics for entity coreference resolution are linguistically agnostic, treating the mentions to be clustered as generic rather than linguistic objects. We argue that the performance of an entity coreference resolver cannot be accurately reflected when it is evaluated using linguistically agnostic metrics. Consequently, we propose a framework for incorporating linguistic awareness into commonly-used coreference evaluation metrics.

## 1 Introduction

Coreference resolution is the task of determining which mentions in a text or dialogue refer to the same real-world entity. Designing appropriate evaluation metrics for coreference resolution is an important and challenging task. Since there is no consensus on which existing coreference evaluation metric is the best, the organizers of the CoNLL-2011 and CoNLL-2012 shared tasks on unrestricted coreference (Pradhan et al., 2011, 2012) decided to take the average of the scores computed by three coreference evaluation metrics, MUC (Vilain et al., 1995), B<sup>3</sup> (Bagga and Baldwin, 1998), and CEAF<sub>e</sub> (Luo, 2005), as the official score of a participating coreference resolver.

One weakness shared by virtually all existing coreference evaluation metrics is that they are *linguistically agnostic*, treating the mentions to be clustered as generic rather than linguistic objects. In other words, while MUC, B<sup>3</sup>, and CEAF were designed for evaluating coreference resolvers, their linguistic agnosticity implies that they can be used to evaluate *any* clustering task, including those that are not linguistic in nature.<sup>1</sup>

<sup>1</sup>This statement is also true for BLANC (Recasens and Hovy, 2011), a Rand Index-based coreference evaluation metric we will not focus on in this paper.

To understand why linguistic agnosticity is a potential weakness of existing scoring metrics, consider a document in which there are three coreferent mentions, *Hillary Clinton*, *she*, and *she*, appearing in this order in the document. Assume that two coreference resolvers,  $R_1$  and  $R_2$ , are applied to these three mentions, where  $R_1$  only posits *Hillary Clinton* and *she* as coreferent, and  $R_2$  only posits the two occurrences of *she* as coreferent. Being linguistically agnostic, existing scoring metrics will assign the *same* score to both resolvers after seeing that both of them correctly assign two of the three objects to the same cluster. Intuitively, however,  $R_1$  should receive a higher score than  $R_2$ :  $R_1$  has facilitated automated text understanding by successfully finding the referent of one of the pronouns, whereas from  $R_2$ 's output we know nothing about the referent of the two pronouns. Failure to rank  $R_1$  higher than  $R_2$  implies that existing scoring metrics fail to adequately reflect the performance of a resolver.<sup>2</sup>

Our goal in this paper is to address the aforementioned weakness by proposing a framework for incorporating linguistic awareness into the most commonly-used coreference scoring metrics, including MUC, B<sup>3</sup>, and CEAF. Rather than making different modifications to different metrics, one of the contributions of our work lies in the proposal of a *unified* framework that enables us to employ the *same* set of modifications to create linguistically aware versions of all these metrics.

## 2 Existing Evaluation Metrics

In this section, we review four scoring metrics, MUC, B<sup>3</sup>, and the two versions of CEAF, namely,

<sup>2</sup>One may disagree that  $R_1$  should be ranked higher than  $R_2$  by arguing that successful identification of two coreferent pronouns is not necessarily easier than resolving an anaphoric pronoun to a non-pronominal antecedent. Our argument, however, is based on the view traditionally adopted in pronoun resolution research that resolving an anaphoric pronoun entails finding a non-pronominal antecedent for it.

CEAF<sub>m</sub> and CEAF<sub>e</sub>. As F-score is always computed as the unweighted harmonic mean of recall and precision, we will only show how recall and precision are computed. Note that unlike previous discussion of these metrics, we present them in a way that reveals their common elements.

## 2.1 Notation and Terminology

In the rest of this paper, we use the terms *coreference chains* and *coreference clusters* interchangeably. For a coreference chain  $C$ , we define  $|C|$  as the number of mentions in  $C$ . *Key chains* and *system chains* refer to gold coreference chains and system-generated coreference chains, respectively. In addition,  $\mathcal{K}(d)$  and  $\mathcal{S}(d)$  refer to the set of gold chains and the set of system-generated chains in document  $d$ , respectively. Specifically,

$$\mathcal{K}(d) = \{K_i : i = 1, 2, \dots, |\mathcal{K}(d)|\},$$

$$\mathcal{S}(d) = \{S_j : j = 1, 2, \dots, |\mathcal{S}(d)|\},$$

where  $K_i$  is a chain in  $\mathcal{K}(d)$  and  $S_j$  is a chain in  $\mathcal{S}(d)$ .  $|\mathcal{K}(d)|$  and  $|\mathcal{S}(d)|$  are the number of chains in  $\mathcal{K}(d)$  and  $\mathcal{S}(d)$ , respectively.

## 2.2 MUC (Vilain et al., 1995)

MUC is a link-based metric. Given a document  $d$ , recall is computed as the number of common links between the key chains and the system chains in  $d$  divided by the number of links in the key chains. Precision is computed as the number of common links divided by the number of links in the system chains. Below we show how to compute (1) the number of common links, (2) the number of key links, and (3) the number of system links.

To compute the number of common links, a partition  $P(S_j)$  is created for each system chain  $S_j$  using the key chains. Specifically,

$$P(S_j) = \{C_j^i : i = 1, 2, \dots, |\mathcal{K}(d)|\} \quad (1)$$

Each subset  $C_j^i$  in  $P(S_j)$  is formed by intersecting  $S_j$  with  $K_i$ . Note that  $|C_j^i| = 0$  if  $S_j$  and  $K_i$  have no mentions in common. Since there are  $|\mathcal{K}(d)| * |\mathcal{S}(d)|$  subsets in total, the number of common links is

$$c(\mathcal{K}(d), \mathcal{S}(d)) = \sum_{j=1}^{|\mathcal{S}(d)|} \sum_{i=1}^{|\mathcal{K}(d)|} w_c(C_j^i), \quad (2)$$

$$\text{where } w_c(C_j^i) = \begin{cases} 0 & \text{if } |C_j^i| = 0; \\ |C_j^i| - 1 & \text{if } |C_j^i| > 0. \end{cases}$$

Intuitively,  $w_c(C_j^i)$  can be interpreted as the “weight” of  $C_j^i$ . In MUC, the weight of a cluster is defined as the *minimum* number of *links* needed to create the cluster, so  $w_c(C_j^i) = |C_j^i| - 1$  if  $|C_j^i| > 0$ .

The number of links in the key chains,  $\mathcal{K}(d)$ , is calculated as:

$$k(\mathcal{K}(d)) = \sum_{i=1}^{|\mathcal{K}(d)|} w_k(K_i), \quad (3)$$

where  $w_k(K_i) = |K_i| - 1$ . The number of links in the system chains,  $s(\mathcal{S}(d))$ , is calculated as:

$$s(\mathcal{S}(d)) = \sum_{j=1}^{|\mathcal{S}(d)|} w_s(S_j), \quad (4)$$

where  $w_s(S_j) = |S_j| - 1$ .

## 2.3 B<sup>3</sup> (Bagga and Baldwin, 1998)

One of MUC’s shortcoming is that it fails to reward successful identification of singleton clusters. To address this weakness,  $B^3$  first computes the recall and precision for each mention, and then averages these per-mention values to obtain the overall recall and precision.

Let  $m_n$  be the  $n$ th mention in document  $d$ . Its recall,  $R(m_n)$ , and precision,  $P(m_n)$ , are computed as follows. Let  $K_i$  and  $S_j$  be the key chain and the system chain that contain  $m_n$ , respectively, and let  $C_j^i$  be the set of mentions appearing in both  $S_j$  and  $K_i$ .

$$R(m_n) = \frac{w_c(C_j^i)}{w_k(K_i)}, P(m_n) = \frac{w_c(C_j^i)}{w_s(S_j)}, \quad (5)$$

where  $w_c(C_j^i) = |C_j^i|$ ,  $w_k(K_i) = |K_i|$ , and  $w_s(S_j) = |S_j|$ .

## 2.4 CEAF (Luo, 2005)

While  $B^3$  addresses the shortcoming of MUC, Luo presents counter-intuitive results produced by  $B^3$ , which it attributes to the fact that  $B^3$  may use a key/system chain more than once when computing recall and precision. To ensure that each key/system chain will be used at most once in the scoring process, his CEAF scoring metric scores a coreference partition by finding an optimal *one-to-one mapping* (or *alignment*) between the chains in  $\mathcal{K}(d)$  and those in  $\mathcal{S}(d)$ .

Since the mapping is one-to-one, not all key chains and system chains will be involved in it. Let

$\mathcal{K}_{min}(d)$  and  $\mathcal{S}_{min}(d)$  be the set of key chains and the set of system chains involved in the alignment, respectively. The alignment can be represented as a one-to-one mapping function  $g$ , where

$$g(K_i) = S_j, K_i \in \mathcal{K}_{min}(d) \text{ and } S_j \in \mathcal{S}_{min}(d).$$

The score of  $g$ ,  $\Phi(g)$ , is defined as

$$\Phi(g) = \sum_{K_i \in \mathcal{K}_{min}(D)} \phi(K_i, g(K_i)),$$

where  $\phi$  is a function that computes the *similarity* between a gold chain and a system chain. The optimal alignment,  $g^*$ , is the alignment whose  $\Phi$  value is the largest among all possible alignments, and can be computed efficiently using the Kuhn-Munkres algorithm (Kuhn, 1955).

Given  $g^*$ , the recall (R) and precision (P) of a system partition can be computed as follows:

$$R = \frac{\Phi(g^*)}{\sum_{i=1}^{|\mathcal{K}(d)|} \phi(K_i, K_i)}, P = \frac{\Phi(g^*)}{\sum_{j=1}^{|\mathcal{S}(d)|} \phi(S_j, S_j)}.$$

As we can see, at the core of CEAF is the similarity function  $\phi$ . Luo defines two different  $\phi$  functions,  $\phi_3$  and  $\phi_4$ :

$$\phi_3(K_i, S_j) = |K_i \cap S_j| = w_c(C_j^i) \quad (6)$$

$$\phi_4(K_i, S_j) = \frac{2|K_i \cap S_j|}{|K_i| + |S_j|} = \frac{2 * w_c(C_j^i)}{w_k(K_i) + w_s(S_j)} \quad (7)$$

$\phi_3$  and  $\phi_4$  result in mention-based CEAF (a.k.a. CEAF<sub>m</sub>) and entity-based CEAF (a.k.a. CEAF<sub>e</sub>), respectively.

## 2.5 Common functions

Recall that the three weight functions,  $w_c$ ,  $w_k$ , and  $w_s$ , are involved in all the scoring metrics we have discussed so far. To summarize:

- $w_c(C_j^i)$  is the weight of the common subset between  $K_i$  and  $S_j$ . For MUC, its value is 0 if  $C_j^i$  is empty and  $|C_j^i| - 1$  otherwise; for B<sup>3</sup>, CEAF<sub>m</sub> and CEAF<sub>e</sub>, its value is  $|C_j^i|$ .
- $w_k(K_i)$  is the weight of key chain  $K_i$ . For MUC, its value is  $|K_i| - 1$ , while for B<sup>3</sup>, CEAF<sub>m</sub> and CEAF<sub>e</sub>, its value is  $|K_i|$ .
- $w_s(S_j)$  is the weight of system chain  $S_j$ . For MUC, its value is  $|S_j| - 1$ , while for B<sup>3</sup>, CEAF<sub>m</sub> and CEAF<sub>e</sub>, its value is  $|S_j|$ .

Next, we will show that simply by redefining these three functions appropriately, we can create linguistically aware versions of MUC, B<sup>3</sup>, CEAF<sub>m</sub>, and CEAF<sub>e</sub>.<sup>3</sup> For convenience, we will refer to their linguistically aware counterparts as LMUC, LB<sup>3</sup>, LCEAF<sub>m</sub>, and LCEAF<sub>e</sub>.<sup>4</sup>

## 3 Incorporating Linguistic Awareness

As mentioned in the introduction, one of the contributions of our work lies in identifying the three weight functions that are common to MUC, B<sup>3</sup>, CEAF<sub>m</sub>, and CEAF<sub>e</sub> (see Section 2.5). To see why these weight functions are important, note that *any interaction between a scoring metric and a coreference chain is mediated by one of these weight functions*. In other words, if these weight functions are linguistically agnostic (i.e., they treat the mentions as generic rather than linguistic objects when assigning weights), the scoring metric that employs them will be linguistically agnostic. On the other hand, if these weight functions are linguistically aware, the scoring metric that employs them will be linguistically aware.

This observation makes it possible for us to design a *unified* framework for incorporating linguistic awareness into existing coreference scoring metrics. Specifically, rather than making different modifications to different scoring metrics to incorporate linguistic awareness, we can simply incorporate linguistic awareness into these three weight functions. So when they are being used in different scoring metrics, we can handily obtain the linguistically aware versions of these metrics.

In the rest of this section, we will suggest one way of implementing linguistic awareness. This is by no means the only way to implement linguistic awareness, but we believe that this is a good starting point, which hopefully will initiate further discussions in the coreference community.

### 3.1 Formalizing Linguistic Awareness

Other than illustrating the notion of linguistic awareness via a simple example in the introduction, we have thus far been vague about what ex-

<sup>3</sup>Note that for a given scoring metric,  $w_c(C) = w_k(C) = w_s(C)$  for any non-empty chain  $C$ . The reason why we define three weight functions as opposed to one is that they are defined differently in the linguistically aware scoring metrics, as we will see.

<sup>4</sup>Our implementation of the linguistically aware evaluation metrics is available from <http://www.hlt.utdallas.edu/~yzcchen/coreference>.

actly it is. In this section, we will make this notion more concrete.

Recall that the goal of (co)reference resolution is to facilitate automated text understanding by finding the referent for each referring expression in a text. Hence, when resolving a mention, a resolver should be rewarded more if the selected antecedent allows the underlying *entity* to be inferred than if it doesn't, because the former contributes more to understanding the corresponding text than the latter. Note that the more *informative* the selected antecedent is, the easier it will be for the reader to infer the underlying entity. Here, we adopt a simple notion of linguistic informativeness based on the mention type: a name is more informative than a nominal, which in turn is more informative than a pronoun.<sup>5</sup> Hence, a coreference link involving a name should be given a higher weight than one that doesn't, and a coreference link involving a nominal should be given a higher weight than one that involves only pronouns.

We implement this observation by assigning to each link  $e_l$  a weight of  $w_l(e_l)$ , where  $w_l(e_l)$  is defined using the first rule applicable to  $e_l$  below:

**Rule 1:** If  $e_l$  involves a name,  $w_l(e_l) = w_{nam}$ .

**Rule 2:** If  $e_l$  involves a nominal,  $w_l(e_l) = w_{nom}$ .

**Rule 3:**  $w_l(e_l) = w_{pro}$ .

There is a caveat, however. By assigning weights to coreference *links* rather than mentions, we will be unable to reward successful identification of singleton clusters, since they contain *no* links (and hence they carry no weights). To address this problem, we introduce a singleton weight  $w_{sing}$ , which will be assigned to any chain that contains exactly one mention.

So far, we have introduced four weights,  $W = (w_{nam}, w_{nom}, w_{pro}, w_{sing})$ , which encode our (somewhat simplistic) notion of linguistic awareness. Below we show how these four weights are incorporated into the three weight functions,  $w_c$ ,  $w_k$ , and  $w_s$ , to create their linguistically aware counterparts,  $w_c^L$ ,  $w_k^L$ , and  $w_s^L$ .

### 3.2 Defining $w_c^L$

Recall that  $C_j^i$  represents the set of mentions common to key chain  $K_i$  and system chain  $S_j$ . To define the linguistically aware weight function  $w_c^L(C_j^i)$ , there are three cases to consider:

<sup>5</sup>Different notions of linguistic informativeness might be appropriate for different natural language applications. In our framework, a different notion of linguistic informativeness can be implemented simply by altering the weight functions.

#### Case 1: $|C_j^i| \geq 2$

Recall that the linguistically agnostic  $w_c$  function returns a weight of  $|C_j^i| - 1$ . This makes sense, because in a linguistically agnostic situation, all the links have the same weight, and hence the weight assigned to  $C_j^i$  will be the same regardless of which  $|C_j^i| - 1$  links in  $C_j^i$  are chosen. However, the same is no longer true in a linguistically aware setting: since the links may not necessarily have the same weight, the weight assigned to  $C_j^i$  depends on which  $|C_j^i| - 1$  links are chosen. In this case, it makes sense for our linguistically aware  $w_c^L$  function to find the  $|C_j^i| - 1$  links that have the largest weights and assign to  $w_c^L$  the sum of these weights, since they reflect how well a resolver managed to find informative antecedents for the mentions. Note that the sum of the  $|C_j^i| - 1$  links that have the largest weights is equal the weight of the maximum spanning tree defined over the mentions in  $C_j^i$ .

#### Case 2: $|C_j^i| = 0$

In this case  $C_j^i$  is empty, meaning that  $K_i$  and  $S_j$  do not have any mention in common.  $w_c^L$  simply returns a weight of 0 when applied to  $C_j^i$ .

#### Case 3: $|C_j^i| = 1$

In this case,  $K_i$  and  $S_j$  have one mention in common. The question, then, is: can we simply return  $w_{sing}$ , the weight associated with a singleton cluster? The answer is no: since  $w_{sing}$  was created to reward *successful* identification of singleton clusters, a resolver should be rewarded by  $w_{sing}$  only if it correctly identifies a singleton cluster. In other words,  $w_c^L$  returns  $w_{sing}$  if all of  $C_j^i$ ,  $K_i$  and  $S_j$  contain exactly one mention (which implies that the singleton cluster  $C_j^i$  is correctly identified); otherwise,  $w_c^L$  returns 0.

The definition of  $w_c^L$  is summarized as follows, where  $E$  is the set of edges in the maximum spanning tree defined over the mentions in  $C_j^i$ .

$$w_c^L(C_j^i) = \begin{cases} \sum_{e_l \in E} w_l(e_l) & \text{if } |C_j^i| > 1; \\ w_{sing} & \text{if } |C_j^i|, |K_i|, |S_j| = 1; \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

### 3.3 Defining $w_k^L$

Recall that  $w_k^L$  aims to compute the weight of key chain  $K_i$ . Given the definition of  $w_c^L$ , in order to ensure that the maximum recall is 1, it is natural to define  $w_k^L$  as follows, where  $E$  is the set of edges

appearing in the maximum spanning tree defined over the mentions in  $K_i$ .

$$w_k^L(K_i) = \begin{cases} \sum_{e_l \in E} w_l(e_l) & \text{if } |K_i| > 1; \\ w_{sing} & \text{if } |K_i| = 1. \end{cases} \quad (9)$$

### 3.4 Defining $w_s^L$

Finally, we define  $w_s^L$ , the function for computing the weight of system chain  $S_j$ . To better understand how we might want to define  $w_s^L$ , recall that in MUC, B<sup>3</sup>, and both versions of CEAF, precision and recall play a symmetric role. In other words, precision is computed by reversing the roles of the key partition  $\mathcal{K}(d)$  and the system partition  $\mathcal{S}(d)$  used to compute recall for document  $d$ . If we wanted precision and recall to also play a symmetric role in the linguistically aware versions of these scoring metrics, it would be natural to define  $w_s^L$  in the same way as  $w_k^L$ , where  $E$  is the set of edges appearing in the maximum spanning tree defined over the mentions in  $S_j$ .

$$w_s^L(S_j) = \begin{cases} \sum_{e_l \in E} w_l(e_l) & \text{if } |S_j| > 1; \\ w_{sing} & \text{if } |S_j| = 1. \end{cases} \quad (10)$$

However, there is a reason why it is undesirable for us to define  $w_s^L$  in this manner. Consider the special case in which a system partition  $\mathcal{S}(d)$  contains only correct links, some of which are suboptimal.<sup>6</sup> Although  $\mathcal{S}(d)$  contains only correct links, the precision computed by any scoring metric that employs  $w_s^L$  with the above definition will be less than one simply because it contains suboptimal links. In other words, if a scoring metric employs  $w_s^L$  with the above definition, it will penalize a resolver for choosing suboptimal links twice, once in recall and once in precision.

To avoid penalizing a resolver for the same mistake twice,  $w_s^L$  cannot be defined in the same way as  $w_k^L$ .<sup>7</sup> In particular, only spurious links (i.e., links between two non-coreferent mentions), not suboptimal links, should be counted as precision errors. To avoid this problem, recall that  $P(S_j)$  is defined as a partition of system chain  $S_j$  created by intersecting  $S_j$  with all key chains in  $\mathcal{K}(d)$ .

$$P(S_j) = \{C_j^i : i = 1, 2, \dots, |\mathcal{K}(d)|\}$$

<sup>6</sup>Suboptimal links are links that are correct but do not appear in a maximum spanning tree for any of its chains.

<sup>7</sup>This implies that precision and recall will no longer play a symmetric role in our linguistically aware scoring metrics.

Note that a link is spurious if it links a mention in  $C_j^{i_1}$  with a mention in  $C_j^{i_2}$ , where  $1 \leq i_1 \neq i_2 \leq \mathcal{K}(d)$ . Without loss of generality, assume that there are  $ne_j$  non-empty clusters in  $P(S_j)$ . Note that we need  $ne_j - 1$  spurious links in order to connect the  $ne_j$  non-empty clusters. To adequately reflect the damage created by these spurious links, among the different sets of  $ne_j - 1$  spurious links that connect the  $ne_j$  non-empty clusters in  $P(S_j)$ , we choose the set where the sum of the weights of the links is the largest and count the edges in it as precision errors. We denote this set as  $E_t(S_j)$ .

Now we are ready to define  $w_s^L$ . There are two cases to consider.

**Case 1:**  $|S_j| > 1$

In this case,  $w_s^L(S_j)$  is computed as follows:

$$w_s^L(S_j) = \sum_{C_j^i \in P(S_j)} w_c^L(C_j^i) + \sum_{e \in E_t(S_j)} w_l(e). \quad (11)$$

Note that the second term corresponds to the precision errors discussed in the previous paragraph, whereas the first term corresponds to the sum of the values returned by  $w_c^L$  when applied to each cluster in  $P(S_j)$ . The first term guarantees that a resolver is penalized for precision errors because of spurious links, not suboptimal links.

**Case 2:**  $|S_j| = 1$

In this case,  $S_j$  only contains one mention. We set  $w_s^L(S_j)$  to  $w_{sing}$ .

## 4 Evaluation

In this section, we design experiments to better understand our linguistically aware metrics (henceforth *LMetrics*). Specifically, our evaluation is driven by two questions. First, given that the *LMetrics* are parameterized by a vector of four weights  $W$ , how do their behaviors change as we alter  $W$ ? Second, how do the *LMetrics* differ from the existing metrics (henceforth *OMetrics*)?

### 4.1 Experimental Setup

We use as our running example the paragraph shown in Figure 1, which is adapted from the Bible domain of the English portion of the OntoNotes v5.0 corpus. There are 19 mentions in the paragraph, each of which is enclosed in parentheses and annotated as  $m_x^y$ , where  $y$  is the ID of the chain to which this mention belongs, and  $x$  is the mention ID.

Figure 2 shows five system responses (a–e) for our running example along with the key chains.

(Jesus)<sub>a</sub><sup>1</sup> came near (Jerusalem)<sub>d</sub><sup>2</sup>. Looking at (the city)<sub>e</sub><sup>2</sup>, (he)<sub>b</sub><sup>1</sup> began to cry for (it)<sub>f</sub><sup>2</sup> and said, (I)<sub>c</sub><sup>1</sup> wish (you)<sub>g</sub><sup>2</sup> knew what would bring (you)<sub>h</sub><sup>2</sup> (peace)<sub>p</sub><sup>4</sup>. But it is hidden from (you)<sub>i</sub><sup>2</sup> (now)<sub>q</sub><sup>5</sup>. (A time)<sub>r</sub><sup>6</sup> is coming when ((your)<sub>j</sub><sup>2</sup> enemies)<sub>n</sub><sup>3</sup> will hold (you)<sub>k</sub><sup>2</sup> in on (all sides)<sub>s</sub><sup>7</sup>. (They)<sub>o</sub><sup>3</sup> will destroy (you)<sub>l</sub><sup>2</sup> and (all (your)<sub>m</sub><sup>2</sup> people)<sub>t</sub><sup>8</sup>.

Figure 1: A paragraph adapted from the Bible domain of the OntoNotes 5.0 corpus.

For conciseness, a mention is denoted by its mention ID, and each connected sub-graph forms one coreference chain. Moreover, the type of a mention is denoted by its shape: a *square* denotes a NAME mention; a *triangle* denotes a NOMINAL mention, and a *circle* denotes a PRONOUN mention. Note that  $S_{you}$ , the set of coreferent “you” mentions consisting of  $\{m_g^2, m_h^2, \dots, m_m^2\}$ , appears in all system responses.

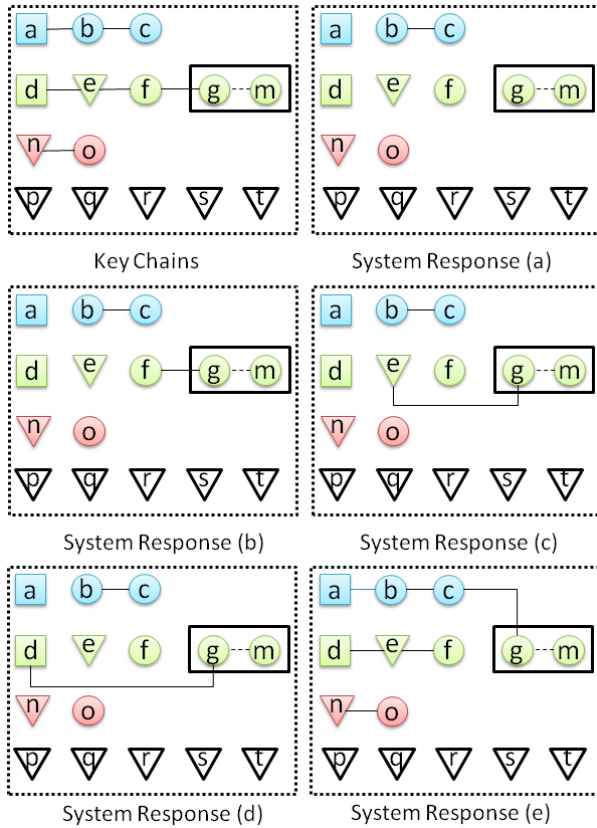


Figure 2: Key and system coreference chains.

Let us begin by describing the five system responses. Response (a) is produced by a simple and conservative resolver. Besides forming  $S_{you}$ , this resolver also correctly links  $m_b^1$  with  $m_c^1$ . Responses (b), (c) and (d) each improves upon response (a) by linking  $S_{you}$  to one of three preceding mentions, namely, one PRONOUN mention, one NOMINAL mention, and one NAME mention respectively. Response (e) is produced by an aggressive resolver that tries to resolve all the pro-

nouns to a non-pronominal antecedent, but unfortunately, it wrongly connects  $S_{you}$  to  $m_a^1$ ,  $m_b^1$  and  $m_c^1$ .

Next, we investigate the two questions posed at the beginning of Section 4.1. To determine how the *LMetrics* behave when used in combination with different weight vectors  $W = (w_{nam}, w_{nom}, w_{pro}, w_{sing})$ , we experiment with:

$$W_1 = (1.0, 1.0, 1.0, 10^{-20});^8$$

$$W_2 = (1.0, 1.0, 1.0, 0.5);$$

$$W_3 = (1.0, 1.0, 1.0, 1.0);$$

$$W_4 = (1.0, 0.75, 0.5, 1.0);$$

$$W_5 = (1.0, 0.5, 0.25, 1.0).$$

Note that  $W_1$ ,  $W_2$ , and  $W_3$  differ only with respect to  $w_{sing}$ , so comparing the results obtained using these weight vectors will reveal the impact of  $w_{sing}$  on the *LMetrics*. On the other hand,  $W_4$  and  $W_5$  differ with respect to the gap of the weights associated with the three types of mentions. Examining the *LMetrics* when they are used in combination with  $W_4$  and  $W_5$  will reveal the difference between having “relatively similar” weights versus having “relatively different” weights on the three mention types.

Figure 3 shows four graphs, one for each of the four *LMetrics*. Each graph contains six curves, five of which correspond to curves generated by using the aforementioned five weight vectors, and the remaining one corresponds to the *OMetric* curve that we include for comparison purposes. Each curve is plotted using five points that correspond to the five system responses.

## 4.2 Impact of $w_{sing}$

We first investigate the impact of  $w_{sing}$ . We will determine how the *LMetrics* behave in response to  $W_1$ ,  $W_2$  and  $W_3$ .

The first graph in Figure 3 shows the LMUC and MUC F-scores. As we can see, the scores of MUC and LMUC( $W_1$ ) are almost the same. This is understandable: the uniform edge weights and a very small  $w_{sing}$  in  $W_1$  imply that LMUC will

<sup>8</sup>We set  $w_{sing}$  to a very small value other than 0, because setting  $w_{sing}$  to 0 may cause the denominator of the expressions in (5) and (7) to be 0.



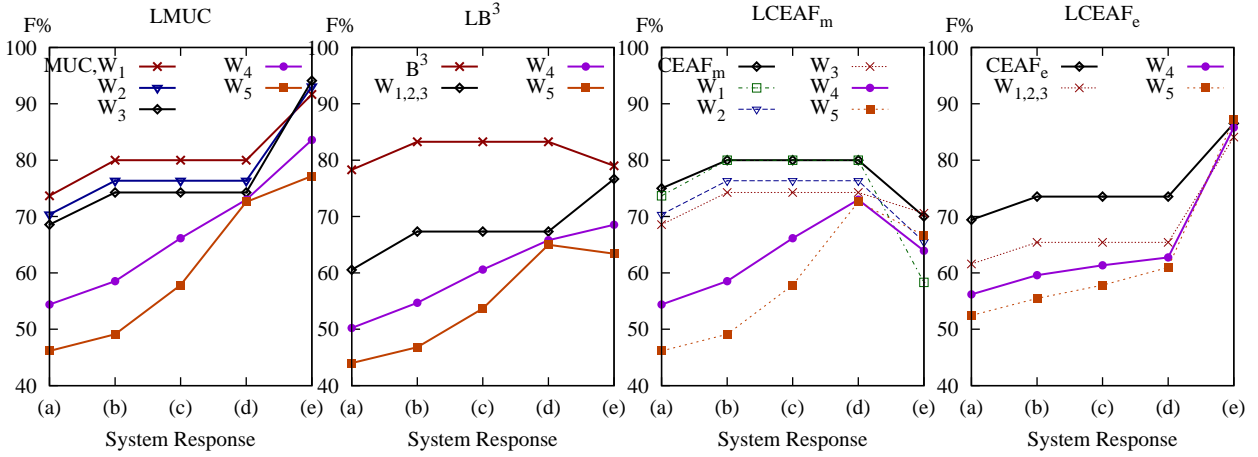


Figure 3: Comparison of the *LMetrics* scores under different weight settings and the *OMetrics* scores.

essentially ignore correct identification of single clusters and consider all errors to be equal, just like MUC. When we replace  $W_1$  with  $W_2$  and  $W_3$ , the two weight vectors with a larger  $w_{sing}$  value, and rescore the five responses, we see that the LMUC scores for responses (a), (b), (c) and (d) decrease. This is because LMUC uses  $w_{sing}$  to penalize these four responses for identifying wrong singleton clusters. On the other hand, the LMUC score for response (e) is higher than the corresponding MUC score, because LMUC additionally rewards response (e) for correctly classifying all singleton clusters without introducing erroneous singleton clusters.

The second graph in Figure 3 shows the  $LB^3$  and  $B^3$  F-scores. Here, we see that the scores for  $LB^3(W_1)$ ,  $LB^3(W_2)$  and  $LB^3(W_3)$  are identical. These results suggest that the value of  $w_{sing}$  does not affect the  $LB^3$  score, despite the fact that  $LB^3$  does take into account singleton clusters when scoring, a property that it inherits from  $B^3$ . The reason is that regardless of what  $w_{sing}$  is, if a mention  $m$  is correctly classified as a singleton mention, both of  $R(m)$  and  $P(m)$  will be 1, otherwise, both will be 0 (see formula (5)). Note, however, that there is a difference between  $LB^3$  and  $B^3$ : for an erroneously identified singleton cluster containing mention  $m$ ,  $LB^3$  sets  $P(m)$  to 0 while  $B^3$  sets  $P(m)$  to 1. In other words,  $LB^3$  puts a higher penalty on precision given erroneous singleton clusters. This difference causes  $LB^3$  and  $B^3$  to evaluate responses (a) and (e) differently. Recall that responses (a) and (e) are quite different: response (e) correctly finds informative antecedents for  $m_b^1, m_c^1, m_e^2, m_f^2$  and  $m_o^3$ , whereas response (a)

contains many erroneous singleton clusters. Despite the large differences in these responses,  $B^3$  only gives 0.7% more points to response (e) than response (a). On the other hand,  $LB^3$  assigns a much lower score to response (a) owing to the numerous erroneous singleton clusters it contains.

The third graph of Figure 3 shows the  $LCEAF_m$  and  $CEAF_m$  F-scores. Since  $LCEAF_m$  uses both singleton and non-singleton clusters when computing the optimal alignment, it should not be surprising that as we increase  $w_{sing}$ , the singleton clusters will play a more important role in the  $LCEAF_m$  score. Consider, for example,  $LCEAF_m(W_1)$ . Since  $w_{sing} = 0$ ,  $LCEAF_m(W_1)$  ignores the correct identification of singleton clusters. From the graph, we see that  $LCEAF_m(W_1)$  gives a higher score to response (a) than response (e). This is understandable: response (a) is not penalized for the many erroneous singleton clusters it contains; on the other hand, response (e) is penalized for the erroneous coreference links it introduces. Now, consider  $LCEAF_m(W_3)$ , where  $w_{sing} = 1$ . Here, response (e) is assigned a higher score by  $LCEAF_m(W_3)$  than response (a): response (a) is heavily penalized because of the many erroneous clusters it contains.

The rightmost graph of Figure 3 shows the  $LCEAF_e$  and  $CEAF_e$  F-scores. Like  $LB^3$ ,  $LCEAF_e$  returns the same score when it is used in combination with  $W_1, W_2$  and  $W_3$ , because the  $\phi_4$  similarity function returns 0 or 1 when the key cluster or the system cluster it is applied to is a singleton cluster, regardless of the value of  $w_{sing}$ . In addition, we can see that  $LCEAF_e$  penalizes erroneous singleton clusters more than  $CEAF_e$  does

chains	MUC			LMUC			B <sup>3</sup>			LB <sup>3</sup>			CEAF <sub>m</sub>			LCEAF <sub>m</sub>			CEAF <sub>e</sub>			LCEAF <sub>e</sub>		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
(a)	58.3	100	73.7	50.7	58.6	54.4	64.3	100	78.3	39.2	70.0	50.2	75.0	75.0	75.0	50.7	58.6	54.4	91.1	56.1	69.4	73.8	45.4	56.2
(b)	66.7	100	80.0	53.7	64.3	58.5	71.3	100	83.3	43.1	75.0	54.7	80.0	80.0	80.0	53.7	64.3	58.5	91.9	61.3	73.6	74.5	49.7	59.6
(c)	66.7	100	80.0	64.2	68.3	66.2	71.3	100	83.3	50.8	75.0	60.6	80.0	80.0	80.0	64.2	68.3	66.2	91.9	61.3	73.6	76.7	51.1	61.4
(d)	66.7	100	80.0	74.6	71.4	73.0	71.3	100	83.3	58.6	75.0	65.8	80.0	80.0	80.0	74.6	71.4	73.0	91.9	61.3	73.6	78.4	52.3	62.8
(e)	91.7	91.7	91.7	76.1	92.7	83.6	79.0	79.0	79.0	65.0	72.5	68.5	70.0	70.0	70.0	58.2	70.9	63.9	86.5	86.5	86.5	85.8	85.8	85.8

Table 1: Comparison of the  $LMetrics(W_4)$  scores and the  $OMetrics$  scores.

for the same reason that  $LB^3$  penalizes erroneous singleton clusters more than  $B^3$  does.

In sum, the value of  $w_{sing}$  does not impact  $LB^3$  and  $LCEAF_e$ . On the other hand,  $LMUC$  and  $LCEAF_m$  pay more attention to singleton clusters as  $w_{sing}$  increases.

### 4.3 Impact of $w_{nam}$ , $w_{nom}$ and $w_{pro}$

When we were analyzing the  $LMetrics$  in the previous subsection, by setting  $w_{nam}$ ,  $w_{nom}$ , and  $w_{pro}$  to the same value, we were not exploiting their capability to be linguistically aware. In this subsection, we investigate the impact of linguistic awareness using  $W_4$  and  $W_5$ , which employ different values for the three weights.<sup>9</sup> To better understand the differences in recall and precision scores for each of the five system responses, we show these scores as computed by the  $LMetrics$  when they are used in combination with  $W_4$ .

First, consider response (a). As we can see from Figure 3 and the first row of Table 1, the  $OMetrics$  give decent scores to this output. Linguistically speaking, however, the system should be penalized more. The reason is that its output contributes little to understanding the document: in response (a), only the links between the PRONOUN mentions are established, and none of the PRONOUN or NOMINAL mentions is linked to a more informative mention that would enable the underlying entity to be inferred.

As expected,  $LMetrics(W_4)$  and  $LMetrics(W_5)$  assign much lower scores to response (a) than the  $OMetrics$ , owing to a relatively small value of  $w_{pro}$ . Also, we see that the  $LMetrics(W_5)$  scores are even lower than the  $LMetrics(W_4)$  scores. This suggests that the smaller the values of  $w_{pro}$  and  $w_{nom}$  are, the more heavily a resolver will be penalized for its failure to link a mention to a more informative coreferent mention.

Next, consider responses (b), (c) and (d). As the

<sup>9</sup>Like  $W_3$ , we set  $w_{sing}$  to 1 in  $W_4$  and  $W_5$ , because this assignment makes  $CEAF_m(W_3)$  rank response (e) above response (a), which we think is reasonable.

$OMetrics$  ignore the type of mentions while scoring, they are unable to distinguish the differences among these three system responses: the  $OMetrics$  results in Figure 3 and their results in rows 2, 3 and 4 of Table 1 show that the scores for responses (b), (c) and (d) are identical. Linguistically speaking, however, they should not be. Response (d) contributes the most to document understanding, because the presence of NAME mention  $m_d^2$  in its output enables one to infer the entity (*Jerusalem*) to which the mentions in  $S_{you}$  refer. In contrast, although response (b) correctly links  $S_{you}$  to PRONOUN mention  $m_f^2$ , one cannot infer the entity to which the mentions in  $S_{you}$  refer. The contribution of response (c) is in-between, because via  $m_e^2$ , we at least know that the mentions in  $S_{you}$  point to one *city*, although we do not know which *city* it is. Such differences in responses (b), (c) and (d) are captured by  $LMetrics(W_4)$  and  $LMetrics(W_5)$ . Specifically, the  $LMetrics$  scores for response (d) are higher than those for response (c), which in turn are higher than those for response (b).

It is worth noting that the performance gaps between responses (b) and (c) and between responses (c) and (d) are larger under  $LMetrics(W_5)$  than under  $LMetrics(W_4)$ . This is because  $w_{nom}$  and  $w_{pro}$  in  $W_5$  are comparatively smaller. These results enable us to conclude that as the difference in the three edge weights becomes larger, the performance gap between a less informative resolver and a more informative resolver according to the  $LMetrics$  widens.

## 5 Conclusion

We addressed the problem of linguistic agnosticity in existing coreference evaluation metrics by proposing a framework that enables linguistic awareness to be incorporated into these metrics. While our experiments were performed on gold mentions, it is important to note that our linguistically aware metrics can be readily combined with, for example, Cai and Strube’s (2010) method, so that they can be applied to system mentions.

## Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF.

## References

- Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the LREC Workshop on Linguistic Coreference*, page 563–566.
- Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. In *Proceedings of the 11th Annual SIGDIAL meeting on Discourse and Dialogue*, pages 28–36.
- Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.
- Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.
- Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning: Shared Task*, pages 1–40.
- Marta Recasens and Eduard Hovy. 2011. BLANC: Implementing the Rand Index for coreference resolution. *Natural Language Engineering*, 17(4):485–510.
- Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference*, pages 45–52.

# An Empirical Assessment of Contemporary Online Media in Ad-Hoc Corpus Creation for Social Events

Kanika Narang, Seema Nagar, Sameep Mehta, L V Subramaniam, Kuntal Dey

IBM Research Labs, India

{kaninara, senagar3, sameepmehta, lvsubram, kuntadey}@in.ibm.com

## Abstract

Social networking sites such as Facebook and Twitter have become favorite portals for users to discuss and express opinions. Research shows that topical discussions around events tend to evolve socially on microblogs. However, sources like Twitter have no explicit discussion thread which will link semantically similar posts. Moreover, the discussion may be evolving in multiple different threads (like Facebook). Researchers have proposed the use of online contemporary documents to act as external corpus to connect pairs of contextually related semantic topics. This motivates the question: *given a significant social event, what is a good choice of external corpus to identify evolution of discussion topics around the event's context?* In this work, we compare the effectiveness of contemporary blog posts, online news media and forum discussions in creating ad-hoc external corpus. Using social propensity of evolution of topical discussions on Twitter to assess the goodness of the creation, we find online news media as most effective. We evaluate on three large real-life Twitter datasets to affirm our findings.

## 1 Introduction

Social media has become a hotbed of user generated content. Multiple online platforms have emerged for users to participate, interact and discuss. Social contact and activity networks like Facebook, video sharing networks like Youtube, photo/image sharing platforms like Pinterest, social bookmarking platforms like Digg, and mi-

croblogging platforms like Twitter have become prominent online social media platforms.

Research suggests that social microblogging platforms, like Twitter, diffuse information in a manner similar to news media (Kwak and Lee, 2010). In a world of millions of people (Twitter subscribers) with inherent entropy, in absence of explicit discussion threads (unlike online forums, for example), conversations around any event are expected to move towards different directions over time. Contradictory to this apparent expectation, research suggests that these discussions tend to temporally grow and evolve along social relationships of people engaging in these discussions, much more strongly, compared to random evolution (Narang and Nagar, 2013).

Interestingly, trending events on unstructured microblogs often get built around non-traditional, temporary and contemporary factors, entities and relationships. Because of the sheer number of diverse contemporary events, event types and associated documents, it is impossible to prepare well-defined, validated and clean corpus for each and every event. For instance, political turmoils have existed for ages; however, one may not expect a dedicated corpus to preexist for the Libya 2011 turmoil associating its places and locations, contemporary leaders, and all the other global political factors. Hence, there is a strong need of using contemporary online media for ad-hoc corpus creation in such a setting.

The use of external corpus has been shown to improve performance in language tasks such as question-answering, machine translation, and information retrieval (Kilgarriff, 2003; Clarke and Cormack, 2002; Dumaisl and Banko, 2002; Metzler and Diaz, 2005; Xu and Croft, 2000; Diaz and Metzler, 2006). But questions relating to the rele-

vance of the corpus have not been studied as much.

The finding of (Narang and Nagar, 2013) that topical discussions on microblogs tend to evolve socially is interesting. However, it simply uses contemporary online news media as the only source of external corpus, to establish *extended semantic relationships* across topic clusters. It does not attempt to use any other source of relevant semantic data for creating external corpus; nor does it assess the goodness of contemporary online news portals for this purpose.

In this study, we propose evaluating the goodness of different sources of external data for constructing ad-hoc corpus to connect topic clusters using extended semantic edges. In addition to the contemporary online news media corpus, we use two other independent external corpus for constructing extended semantic edges, namely contemporary online forum discussions and contemporary blog posts. To the best of our knowledge, ours is the first study of its kind.

We use three large scale real-life Twitter datasets, namely Libya 2011 political turmoil, Egypt 2011 political turmoil and London 2012 Olympics, having thousands of users and up to millions of tweets, to conduct experiments. For all datasets, we find online news media to best capture the evolution of discussion topics along social relationships, measured by the normalized mutual information or *NMI* (Coombs and Dawes, 1970) of the social discussion threads to discussion sequences. We believe this insight to be both novel and interesting. We further observe that, for most cases, online discussion forums perform better compared to blogs.

In summary, the main contributions of our work are the following.

- We empirically evaluate different contemporary relevant external documents, to establish extended semantic relationships across topical clusters formed around events.
- We assess the goodness of contemporary online discussion forums, blogs, online news media and Random search results, in forming extended semantic relationships, and find online news media to be most effective in creating ad-hoc corpus around given events.
- We demonstrate our findings on microblogging data using three real events.

## 2 Problem settings and our approach

### Problem settings

We observe the following:

1. There exist several concepts that are connected in a given context, but are not connected by any widely accepted relationship such as synonyms, antonyms, hypernymns, hyponymns *etc.* when taken in isolation. As an example, *damage* and *relief* are intuitively connected concepts in semantic clusters containing  $\{damage, fire, death, toll\}$  and  $\{fire, relief, spray, water\}$ . Yet, none of the traditional semantic relationships will connect these when considered in absence of the larger context, namely an event of fire. Practically, discussions on microblogs about damage caused by fire stands a realistic chance to evolve towards discussions about relief.

2. Events on microblog networks form around non-traditional, temporary and contemporary factors, entities and relationships. It is impractical to expect well-defined corpus to exist a priori.

Clearly, creating corpus applicable for a given event, to be able to connect concepts that are related in context of the event, is a research problem to solve. It is also important to assess the quality of the corpus created in the process.

### Algorithm

In absence of traditional a priori corpus, we attempt to construct ad-hoc corpus applicable to the context of the event. We follow the approach of (Narang and Nagar, 2013) to construct our graphs, conducting our experiments and measuring the goodness of our results. We use Twitter as our testbed. We attempt to use four independent types of contemporary external documents to be able to connect concepts related contextually, namely online forum discussions, blog posts, news media and Random search documents, to derive the *extended semantic relationships*. Our approach consists of the following steps.

#### 2.1 Topic-based cluster creation

We collect tweets belonging to an event from Twitter for our experiments. The whole tweet corpus is divided into clusters of tweets which are semantically related. Event topic cluster detection not being the focus of our work, we use an existing online clustering algorithm (Weng and Lee, 2011) to create clusters of topics related semantically. A semantic event cluster  $E^i$  is represented

as  $\{K^i, [T_s^i, T_e^i]\}$ , where  $K^i$  denotes the set of keywords extracted from the tweets which form the event  $E^i$  and  $T^i$  is time period of the event. We use existing established methods for computing K and T. K contains *idf* vector and proper nouns (extracted by PoS tagging) from the tweets, and uses Stanford's NLP Toolkit and the associated Named Entity Recognizer. T is simply the time of first and last tweet in the event cluster.

## 2.2 Extracting the relationships

Essentially, we generate an event topic graph  $\mathcal{G} = \{\mathcal{E}, \{R\}\}$ , in which  $\mathcal{E}$  represents the event topics (topical clusters), and act as the vertices of the graph. The set  $\{R\}$  represents the relationship edges between the clusters, and are formed from each of contextual semantic, temporal and social perspectives. We, hereby, elaborate on the algorithm used for extracting these relationships.

**Extended Semantic Relationships:** This relationship is extremely useful but challenging to establish. Lets us motivate the need for such relationship by a simple example. Consider two events with associated keywords  $E_1 = \{\text{damage, earthquake, dead, toll}\}$  and  $E_2 = \{\text{earthquake, relief, shelter}\}$ . Now, lets pick one work from each set *damage* and *relief*. One cannot establish any of the widely accepted relationships like synonym, antonyms, hypernyms, hyposynms etc when the words are taken in isolation. However, coupled with prior knowledge about the larger event *earthquake*, the words can be semantically related. In essence (with abuse of notation and terminology), damage and relief are independent variables without extra information, however, they are related given *earthquake*. Therefore, we would like to add the semantic edge between these events. We use external corpus to extract and quantify such semantic relationships. (Narang and Nagar, 2013) used only Google News for creation of the external corpus. But, due to the increased presence of users on Internet, these global and prominent topics are bound to be discussed in blogs and online discussions forums. The natural question which arise is then, which corpus is deemed to be best for the purpose. In this paper, we use different data sources as ad-hoc corpus, namely contemporary online discussions, blogs, online news and Random Search documents, to form four different graphs extended semantic per event. The novelty of our work lies in empirically determining

the goodness of each of these four different data source types in forming ad-hoc corpus.

### Extended semantic relationship extraction

We establish weighted extended semantic relationships across event clusters by the following steps. The input to the extended semantic relationship extraction algorithm for two events  $E^i$  and  $E^j$  is keyword list  $K^i$  and  $K^j$ .

#### Step 1: Generating Pairs and Pruning

**Mechanism-** We generate  $|K^i| \times |K^j|$  pairs of keywords which need to be evaluated for extended semantic relationship. Such large number of pairs would pose computational issues. To handle this, we prune pairs which are related semantically (synonyms, antonyms, hypernyms and hyponyms). We look at the similarity scores of  $K^i$  and  $K^j$  in Wordnet. We use the well-established Lin's method (Lin, 1998) to compute similarity scores of  $K^i$  and  $K^j$  using the feature vector built into the Wordnet lexical database. For sake of completeness, please note that Lin's measure of similarity between pair of words w1 and w2 is defined as:

$$sim(w1; w2) = \frac{2I(F(w1) \cap F(w2))}{I(F(w1)) + I(F(w2))}$$

, where  $F(w)$  is the set of features of a word w, and  $I(S)$  is the amount of information contained in a set of features S. Assuming that features are independent of one another,  $I(S) = -\sum_{f \in S} P \log(P(f))$ , where  $P(f)$  is the probability of feature f.

We retain a pair of words if the similarity score  $S_{ij}$  is lesser than a desirable similarity threshold  $S$ , and discard the pair otherwise. Since POS tagging is done on the tweets in the event, we also remove pairs where one of the word is Proper Noun or Active Verbs.

#### Step 2: Document Corpus Generation and Searching-

We use the keywords used for filtering Twitter Public API to search for news stories for the same time period on contemporary external documents. The retrieved documents act as our external corpus. We create an inverted index for this corpus, where for each word we store the document ids as well as the frequency of the word in the documents. Given the pair of words  $(K_l^i, K_m^j)$  (we will omit subscript l and m, when there is no ambiguity), we find the intersection of corresponding document lists. Therefore, at the end of this step we have list of documents (denoted by  $D_{lm}$ ) in which both the

words co-occur along with their frequency in the documents.

**Step 3: Pairwise Score Computation** - For each of the selected document, we compute the coupling of the pair of words. Assume,  $C(K_l^i, D_t)$  gives the *tf-idf* score of word  $K^i$  in document  $D_t$ . The pairwise coupling can be computed as minimum  $(C(K_l^i, D_t), C(K_m^j, D_t))$ . The overall coupling is calculated as average of coupling over all documents.

**Step 4: Overall Score Computation** - This process is repeated for all pair of words in  $E^i$  and  $E^j$ . Finally, for a given pair of event clusters  $E^i$  and  $E^j$ , if  $w_{ij}$  words were discarded and the rest were retained, then

$$overall\_score = \frac{\sum_{K^i, K^j} Coupling}{(|K^i| \times |K^j| - w_{ij})}.$$

The final scores are ranked in descending order and top K% are selected based on user preference or can be pruned based on threshold.

**Social Relationships:** Direct social connections are the core constituent elements of social relationships. Higher order social relationships can be established by exploring the social network structure. Well-defined structures such as communities with maximum modularity (Girvan and Newman, 2002; Clauset and Newman, 2004) can be extracted using efficient modularity maximization algorithms such as BGLL (Blondel and Guillaume, 2008).

#### Social relationship extraction

We construct social linkage graphs between pairs of events using social connections of event cluster members to construct edges. Each event associates a number of microblog posts (tweets) from a set of members of the microblog network (Twitter).

A person  $P$  is said to belong to an event cluster  $E^i$  iff  $(\exists M)$ , a microblog post, made by  $P$ , such that  $M \in E^i$ . Please note that with this definition, a person can potentially belong to multiple event clusters at the same time.

These connections are established by participation of direct social neighbors of individuals across multiple events. We draw an edge across a given pair of events if there is at least one direct (one-hop) neighbor in each event belonging to the pair of events. The weight of an edge between event cluster  $E^i$  and  $E^j$  is determined by the total number of one-hop neighbors existing between

these two clusters. So if  $E^i$  has  $P^i$  memberships,  $E^j$  has  $P^j$  memberships, the average number of neighbors in  $E^j$  of a member belonging to  $E^i$  is  $a_{ij}$  and the average number of neighbors in  $E^i$  of a member belonging to  $E^j$  is  $a_{ji}$  then the strength of the social edge between  $E^i$  and  $E^j$  is  $(P^i.a_{ij} + P^j.a_{ji})$ .

**Temporal relationship** The third kind of relationship we extracted is temporal relationship. We look at two kinds of temporal relationships. (a) We draw a temporal edge from event  $E^i$  to event  $E^j$  if  $E^i$  ended before  $E^j$  started and the timespan between the two events has to be less than or equal to 2 days. This follows from the assumption that on microblogging services like Twitter, a discussion thread will not last longer than this. This thresholding also prevents the occurrence of spurious edges across different clusters. It captures the *meets* and *disjoint* relationships described by (Allen and J.F., 1983). We call this a  $T_1$  temporal relationship. (b) We draw a temporal edge from event  $E^i$  to event  $E^j$  if  $E^i$  started before  $E^j$  started, and ended after the start but before the end of  $E^j$ . This captures the *overlaps* relationship described by (Allen and J.F., 1983). We call this a  $T_2$  temporal relationship. Please note that unlike the undirected semantic and social relationship edges, a temporal relationship edge is always directed. The source of a temporal relationship edge is the event with the earlier starting time, and the sink is the one with the later starting time.

### 2.3 Identifying and characterizing discussions

Finally, after establishing the relationships, we identify Discussion and Social discussion sequences in the same manner as described by (Narang and Nagar, 2013).

**Identifying discussion sequences:** A discussion sequence graph is defined as, a directed acyclic graph (DAG) of topics that are related using the semantic edges obtained by our earlier semantic relationship extraction process, where the relationships are established over time. Intuitively, a discussion sequence captures the topical evolution of discussions over time. We identify discussion sequences using the logical intersection (AND) of the relationship set of the undirected semantic and the directed temporal graphs, with the directions of the latter preserved in the output.

So, the discussion sequence DAG  $\mathcal{G}_{DS}$  is formed as:  $\mathcal{G}_{DS} = \{\mathcal{E}, \{\{R_T\} \cap \{R_S\}\}\}$ , where the set  $\{R_T\}$  represents the edge set of the directed temporal graph and the set  $\{R_S\}$  represents the edge set of the undirected semantic graph.

**Identifying Social discussion sequences:** We take the above graph, and take an edge set intersection with the social graph. This results in retaining the discussion sequences that are socially connected and eliminating the discussion sequences that are socially disconnected. The retained discussion sequences show the social evolution of discussion topics around events on microblogs. Hence, these socially connected discussion sequences are identified as social discussion threads.

## 2.4 Evaluation

In order to measure the goodness of the approach, we find the BGLL (Blondel and Guillaume, 2008) communities for the discussion sequence graphs and social discussion threads, and compute the normalized mutual information (NMI) (Coombs and Dawes, 1970) for each of these intersections. Please note that NMI values range between 0 and 1, and higher NMI values indicate higher overlaps of the two inputs.

(Narang and Nagar, 2013) showed that Discussion threads tend to evolve socially and as a result, the NMI values between communities formed on Discussion Sequences and Social Discussion thread is higher than in between BGLL communities formed on purely Social and Semantic Graph. In this paper, we will compare the NMI values between Discussion threads and Social Discussion threads with taking different extended semantic graphs for their construction. The corpus which results in highest NMI value between the two graphs has most relevant retrieved documents for the event.

## 3 Results

We collect Twitter data from three events that had created significant impact on social media - Libya 2011 political turmoil (collected 4 - 24 Mar'11), Egypt 2011 political turmoil (collected 1 - 4 Mar'11) and the London 2012 Olympics (collected 27 Jun - 13 Aug'12). We use Google News (<http://news.google.com>) with custom date ranges to collect the contemporary online news

data, and Google blog and discussions search options on Google's portal (<http://w.google.com>) with custom date ranges to collect the blog and forum discussions data respectively. We also used Google Search (<http://google.com>) to collect random search results for the same events which will be a mixture of all the data sources to act as a baseline. We gave the same keywords over the same time range while collecting documents from Google which were used for collection of tweets in the Twitter. Table 1 shows the basic statistics of the datasets.

Following the approach outlined in Section 2, we form semantic topic clusters from the tweets following the algorithm of (Weng and Lee, 2011). We now establish extended semantic, social and temporal relationships. For extended semantic relationships, we form four graphs, one each for online news media, discussions, blog and Random documents. For temporal relationships, we form two graphs, one each for the *follows* and *overlaps* relationships. Thus overall, for each dataset (Libya, Egypt and London), we construct 8 different graphs, constructing a total of 24 graphs for experimentation.

We now identify the discussion sequences by taking a logical intersection of the extended semantic and temporal graphs, and the social discussion threads by taking a logical intersection of the discussion sequences with the social relationship graph. We find the NMI (Coombs and Dawes, 1970) across these two graphs using the BGLL (Blondel and Guillaume, 2008) communities formed around these two graphs. We retain the top 10%, 20%, 30%, 40% and 50% of the graph edges and repeat our experiments to observe the overall trend. Figure 1 captures our findings for the temporal *follows* relationships.

The results clearly indicate that in each case, contemporary online news yields the best results (maximum NMI values). In most cases (except Egypt), online discussion forums give better results compared to blogs. This trend becomes more yet prominent as we retain higher fraction of the relationships. Random search results generally behave the worst, except in London which is a little surprising and interesting.

Table 2 shows the corresponding results for the temporal *overlaps* relationship for Libya, which also prominently shows a similar trend. We observe similar trends for other temporal *over-*



Table 1: Keywords used to collect the Twitter datasets, dates of data collection, number of users, tweets collected and clusters, and number of contemporary external news, forum and blogs documents collected

Dataset	Keywords	NumUsers	Tweets	Clusters	#News	#Forum	#Blogs
Libya	Libya, Gaddafi	83,177	1,011,716	1,344	3,266	280	263
Olympics	London, Olympic	1,313,578	2,319,519	299	1,186	516	307
Egypt	Egypt, Protest	37,961	60,948	141	1,753	513	285

Table 2: NMI values for temporal *overlaps* based graphs of Libya

Source	10%	20%	30%	40%	50%
Blogs	0.01	0.04	0.09	0.09	0.13
Forums	0.01	0.06	0.06	0.11	0.16
News	0.02	0.09	0.14	0.09	0.17
Random	0.02	0.04	0.00	0.08	0.11

*laps* graphs also, with the London Olympics data shows a few exceptions (omitted for space constraints).

To eliminate the possible bias due to the number of documents received for each type of corpus, we also repeated the experiment with taking top 200 documents from each sources namely, Online news, blogs, discussions and Random documents. We, then evaluated the performance of these corpora on the Libya dataset.

The figure 2 shows the NMI graph for the Libya dataset with taking only top 200 articles from each of the data sources. The contemporary news article consistently give best results even in this case which corresponds to the finding in the above experiment. Although, Random results also give an equivalent performance but this can be attributed to the fact that the initial results in Google search mostly contains Google News results which will be in prominence because of the low number of documents selected in this experiment.

## 4 Related Work

Significant research has been conducted on content analysis of information discussed on social media sites (Kwak and Lee, 2010). Grinev et al. (Grinev and Grineva, 2009) demonstrate Tweet-Sieve, a system that obtains news on any given subject by sifting through the Twitter stream. Along similar lines, Twinner by Abrol et al. Abrol and Khan (Abrol and Khan, 2010) identify news content of a query by taking into account the geographic location and the time of query. Nagar et al.

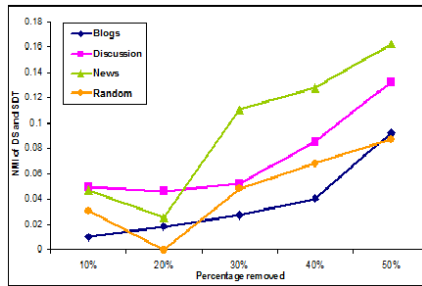
(Nagar and Seth, 2013) demonstrate how content flow occurs during natural disasters.

Several ways to cluster social content have been studied. There has been work on clustering based on links between the users by doing agglomerative clustering, min-cut based graph partitioning, centrality based and Clique percolation methods ((Porter and Onnela, 2009), (Fortunato, 2007)). Other approaches consider only the semantic content of the social interactions for the clustering (Zhou, 2006). More recently there has been work on combining both the links and the content for doing the clustering ((Pathak and Delong, 2008), (Sachan and Contractor, 2012)). In (Narang and Nagar, 2013) relationships between clusters are determined based on semantic, social and temporal information but did not study the impact of different corpus on their results.

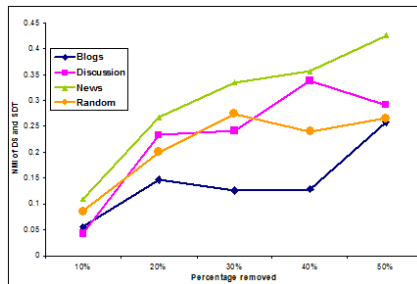
External corpora have been used by researchers to create knowledge base in various fields like for question-answering (Clarke and Cormack, 2002; Dumaisl and Banko, 2002) models such as Chatbots etc, helping machine to translate documents like expanding queries (Kilgarriff, 2003; Metzler and Diaz, 2005) and also for improving Information retrieval using external information (Xu and Croft, 2000; Diaz and Metzler, 2006). They use generic corpora and to the best of our knowledge, there is no study which analyses the relevance of different corpora for the given problem.

## 5 Conclusions

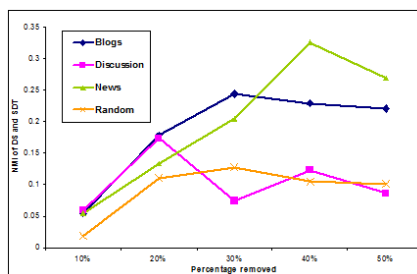
In this work, we studied different contemporary online external data sources for constructing ad-hoc corpus to connect event topic clusters. We explored the content of contemporary online discussion forums, blogs, online news media and Mixture of different corpus, and evaluated their effectiveness in establishing semantic relationships across topical clusters. Exploiting the social propensity of evolution of such discussions, we assessed the goodness of these diverse data sources



(a) NMI for Libya turmoil



(b) NMI for London Olympics



(c) NMI for Egypt turmoil

Figure 1: NMI of social discussion threads (SDT) with respect to discussion sequences (DS): temporal *follows* relationship

using Twitter as a microblogging platform, and eventual NMI values as a qualitative indicator of the goodness of the extended semantic relationships established.

We found contemporary online news media to be the most effective type of external data source for creating ad-hoc corpus, using three large real-life Twitter datasets collected around major events. Further, we found contemporary online discussion forums to be usually, but not always, more effective compared to contemporary blogs. We also found using Mixture of all documents to be mostly give the worst performance.

Our work will be useful to studies and applications that require capturing the evolution of topical discussions on microblogs like Twitter. As future work, we propose evaluating other external

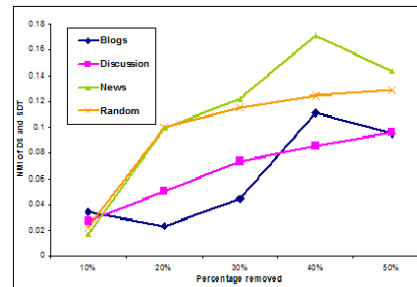


Figure 2: NMI of social discussion threads (SDT) with respect to discussion sequences (DS): temporal *follows* relationship with retaining only top 200 articles

sources of semantic data, and also apply on other microblogging platforms and data sets, for a more comprehensive and complete study.

## References

- S. Abrol and L. Khan. Twinner: Understanding news queries with geo-content using twitter. In *Proceedings of the GIS*, 2010.
- Allen J. F.: *Maintaining Knowledge about Temporal Intervals*. In: Communications of the ACM (1983).
- Blondel V.D., Guillaume J.L., Lambiotte R., Lefebvre E.: *Fast unfolding of communities in large networks*. In: J. Stat. Mech. P10008 (2008).
- Clarke C. L. A., Cormack G. V., Laszlo M., Lynam T. R., Terra E. L.: *The impact of corpus size on question answering performance*. In: SIGIR 02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (2002).
- M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proc. Natl. Acad. Sci, USA*, 99(7821),2002.
- Clauset A., Newman M. E. J., Moore C.: *Finding community structure in very large networks*. In: Phys. Rev. E. 70(066111) (2004).
- Coombs C. H., Dawes R. M., Tversky A.: *Mathematical psychology: An elementary introduction* In: Englewood Cliffs, NJ: Prentice-Hall (1970).
- Diaz F., Metzler D.: *Improving the estimation of relevance models using large external corpora*. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.
- Dumais S., Banko M., Brill E., Lin J., Ng. A. : *Web question answering: is more always better?* In SIGIR 02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (2002).

- S. Fortunato and M. Barthelemy. Resolution limit in community detection. In *Proceedings of the National Academy of Sciences*, 104(1):3641, 2007.
- M. Grinev, M. Grineva, A. Boldakov, L. Novak, A. Syssoev and D. Lizorkin. Tweetsieve: Sifting microblogging stream for events of user interest. In *Proceedings of the SIGIR*, 2009.
- Kilgarriff A., Grefenstette. G.: *Introduction to the special issue on the web as corpus*. In: *Computational linguistics* 29.3 (2003): 333-347.
- Kwak H., Lee C., Park H., Moon S.: *What is Twitter, a Social Media or a News Media*. In: *Proceedings of the WWW (2010)*. In: *Proceedings of the WWW (2010)*.
- Lin D.: *An information-theoretic definition of similarity*. In: *Proceedings of the International Conference on Machine Learning (1998)*.
- Metzler D., Diaz F., Strohman T., Croft W. B.: *Umass at robust 2005: Using mixtures of relevance models for query expansion*. In: *The Fourteenth Text REtrieval Conference (TREC 2005) Notebook (2005)*.
- Narang K., Nagar S., Mehta S., Subramaniam L.V., Dey K.: *Discovery and analysis of evolving topical social discussions on unstructured microblogs*. In: *European Conference on Information Retrieval (2013)*.
- S. Nagar, A. Seth and A. Joshi. Characterization of Social Media Response to Natural Disasters. In *Proceedings of the WWW*, 2012.
- N. Pathak, C. DeLong, A. Banerjee and K. Erickson. Social topics models for community extraction. In *Proceedings of the 2nd SNA-KDD Workshop*, 2008.
- M. A. Porter, J.-P. Onnela and P. J. Mucha. Communities in networks. In *Notices of the American Mathematical Society*, 56(9):1082-1097, 2009.
- M. Sachan, D. Contractor, T. A. Faruque and L. V. Subramaniam. Using Content and Interactions for Discovering Communities in Social Networks. In *Proceedings of the WWW*, 2012.
- Weng J., Lee B.S: *Event detection in Twitter*. In: *IC-SWM - Proceedings of the AAAI conference on weblogs and social media (2011)*.
- Xu J., Croft. W. B.: *Improving the effectiveness of information retrieval with local context analysis*. In: *ACM Trans. Inf. Syst.*, 18(1):79-112 (2000).
- D. Zhou, E. Manavoglu, J. Li, C. L. Giles and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of the WWW*, 2006.

# Diagnosing Causes of Reading Difficulty using Bayesian Networks

**Pascual Martínez-Gómez**

The University of Tokyo  
National Institute of Informatics  
pascual@nii.ac.jp

**Akiko Aizawa**

The University of Tokyo  
National Institute of Informatics  
aizawa@nii.ac.jp

## Abstract

There is a need of matching text difficulty to the expected reading skill of the audience. Readability measures were developed with this objective in mind, first by psycholinguists, and more recently, by practitioners of natural language processing. A common strategy was to extract linguistic features that are good predictors of readability, and then train discriminative classification or regression models that correlate well with human judgment. But correlation does not imply causality, which is a necessary property to explain why documents are not readable. Our objective is to provide mechanisms for text producers to adjust the readability of their content. We propose the use of generative models to diagnose causes of reading difficulty, and bring closer the realization of automatic readability optimization.

## 1 Introduction

Educational institutions, government agencies and some private companies have a special interest in authoring documents for a certain audience, but it is expensive to involve expert linguists to assess the readability of every document they produce. The first psycholinguistic studies developed readability formulas for grading purposes, based on surface linguistic features. Those formulas, despite of their simplicity, performed well and were widely used by editors to grade reading material for young readers. However, content producers might be tempted to adapt their manuscripts by tweaking the text features present in readability formulas, without gaining (or even degrading) real readability (Davison and Kantor, 1982).

Recently, the application of statistical models to linguistic problems proved successful, and am-

bitious tasks in automatic document transformation such as text summarization or machine translation became a hot topic in computational linguistics. Readability optimization is one of such document transformation problems. Recent studies on readability embrace machine learning techniques to recognize readability with an even higher accuracy. The common approach consists in extracting as many features as possible, and then training a classifier or a regression model using human annotated texts to predict a readability score given the observation of the linguistic features.

Those discriminative models correlate well with human judgment, but fail at explaining why a document is not readable. We call *readability diagnosis* the automatic discovery of the causes that lead to (un)readability, and we believe it is an essential step for readability optimization.

We propose the use of Bayesian causal networks to perform readability diagnosis. That is, given a document, the objective of our Bayesian network is to recognize the specific parts of the document that are difficult to read. Bayesian networks are a type of generative model, where the joint probability distribution is constructed by making certain independence assumptions. Their main advantage is that they allow to query the model regarding any linguistic variable, generalizing the functionality of traditional models.

In the next section we briefly introduce former work by psycholinguists and recent work by practitioners on natural language processing. We describe our application of Bayesian networks to readability diagnosis in Section 3 and summarize the capabilities of the model. Corpora, baseline system description and results are presented in Section 4, where we assess to what extent our generative model predicts cognitive evidence. Pointers to future work and applications that would benefit from our results can be found in Section 5, followed by our conclusions in Section 6.

## 2 Related Work

Readability formulas have been the subject of investigation long before the existence of current natural language processing techniques. Although sophisticated methods could have been developed, there was an emphasis on easy-to-compute formulas, where the readability score of a text is computed as a function of its linguistic features.

The Flesch-Kincaid formula (Flesch, 1948; Kincaid et al., 1975) was probably the first in gaining wide recognition among publishers. This formula is a linear combination of two variables<sup>1</sup>, as:

$$r_{\text{score}} = 0.39 \times \text{ASL} + 11.8 \times \text{ASW} - 15.59 \quad (1)$$

where ASL is the average number of words per sentence, and ASW is the average number of syllables per word. Despite of its simplicity, ASL and ASW are very discriminative linguistic features when assessing readability and this formula correlates surprisingly well with human judgment.

The search for more discriminative linguistic features continued, and Mc Laughlin (1969) found that the number of polysyllabic words in a certain amount of text is also a good predictor of reading difficulty. The rationale behind such a linguistic feature could be that the required lexical processing is higher when the word is longer, or that long words tend to be more infrequent and difficult to read (Rayner and Duffy, 1986). This work was followed by others (Fry, 1990) that counted the number of words in the text that were contained in the vocabulary of specific word lists. The use of word lists introduced a new dimension in readability, since it was possible to design hand-crafted lists that could not only account for lexical frequency, but also for semantic complexity.

Building on the idea of lexical frequency and counting on large amounts of text data, the use of word lists was generalized into unigram language models (Si and Callan, 2001), which increased the correlation with human judgment on readability.

Linguistic features of different nature were also explored, and grammatical features are an example of them (Heilman et al., 2007). Those features alone were found not to be as discriminant as the lexical ones, but performed well in combination with them. However, the effects were not additive, which suggests that variables correlated with each

---

<sup>1</sup>We will use the term *variables* interchangeably with *linguistic features*.

other to a certain extent. This was also noted in some other works (Petersen and Ostendorf, 2009), where syntactic features were also used, in combination with higher order n-gram language models.

Automatic text transformation for readability optimization is a task that naturally follows former readability studies and the large scale need of producing content for specific audiences. Authors in (Carroll et al., 1999; Devlin et al., 1999; Siddharthan, 2003) approached the problem using rules for syntactic transformation, anaphora substitutions and vocabulary simplifications, but those rules were not experimentally tested for their target readers. Williams and Reiter (2005) did test their transformation rules, but they were limited to assess the effects of their set of rules, which had a low coverage. Other authors (Aluísio et al., 2010; François and Fairon, 2012) integrated readability scores in authoring systems, to assist text simplification rather than fully automating it.

Previous work has concentrated on finding linguistic features that are good predictors of readability, and building discriminative models that best correlate with human judgment. But those models can only indicate whether a piece of text is readable or not, and fail in explaining the causes. In view of (semi-)automatic text simplification and readability optimization, we propose Bayesian causal networks as a generative model for readability. In this approach, readability is modeled as a factored joint probability distribution over lexical, part of speech, syntactic, semantic and discourse features. This provides an interpretable model to gain linguistic insight about what features impact most on readability in a *specific document* and to understand how that text should be transformed to increase readability even under human-imposed constraints.

## 3 Methodology

### 3.1 Discriminative and Generative Models

Previous work on readability assessment has focused on the development of discriminative models. Those discriminative models are functions  $\phi$  that map instantiations  $\ell$  of a set of linguistic features  $\mathcal{L}$  to a readability score  $r \in \mathbb{R}$ ,  $\phi : \mathcal{L} \rightarrow \mathbb{R}$ . In this work, examples of linguistic features are “proportion of verbs to words”, or “maximum number of active lexical chains” in a given text, and their instantiations are their actual values for that text. If we normalize the readability measure

so that it assigns 1 to the whole space of possible feature instantiations, we can regard the readability score as a probabilistic measure, and without loss of generalization, reformulate the problem as:

$$\hat{r} = \underset{r}{\operatorname{argmax}} \operatorname{Pr}(r \mid \ell), \quad (2)$$

where we have to find the readability score  $r$  with maximum probability, given the instantiation  $\ell$  of the set of linguistic features  $\mathcal{L}$ .

In this approach, the probability on all possible reading score assignments is well defined, but there is no attempt to model the probability of the instantiations  $\ell$  of linguistic features  $\mathcal{L}$ . As it has been reported in related work, most explanatory effects on readability do not add up across all linguistic features. This suggests that linguistic features interact with each other and have mixed effects on readability prediction. There have been ablation and correlation studies to bring light on those feature interactions (Kate et al., 2010), but they were limited to a few feature combinations and no attempts were done to study causal relationships or other conditional independencies.

To attain diagnosis capabilities, we propose the use of Bayesian causal networks as an example of generative models  $\operatorname{Pr}(r, \ell)$ , where the readability score and the linguistic features are modeled *together* using a joint probability distribution. There are, however, some challenges associated to this model that are described below.

### 3.2 Independence Assumptions

To preserve generality, we will regard joint probability distributions as tables, where every row defines the probability of a discrete value assignment to all linguistic features and the readability score. The number of parameters to be estimated in the model is proportional to the number of possible assignments, which is exponential with the number of linguistic features. A simple approach is to consider the readability to be dependent on all features, but all features independent from each other. The joint probability distribution can be consequently defined as:

$$\begin{aligned} \operatorname{Pr}(r, \mathcal{L}) &= \operatorname{Pr}(r, l_1, \dots, l_m) \\ &\approx p(r \mid l_1, \dots, l_m) \cdot p(l_1) \cdots p(l_m), \end{aligned} \quad (3)$$

where  $p(l_i)$  are the priors for every linguistic feature  $l_i$ , and the conditional probability distribution

$p(r \mid l_1, \dots, l_m)$  models the non-linear relationship between linguistic features and the readability score. The graphical representation of this model can be found in Figure 1a, where the gray circles are observed linguistic features, and edges encode probabilistic influence (or dependency). Due to the simplicity of this network, the number of dependencies in  $p(r \mid l_1, \dots, l_m)$  is large, which requires to estimate millions of parameters if there are more than twenty linguistic features.

In order to reduce the number of parameters without reducing the number of linguistic features, we will introduce language constructs in the form of hidden variables and set structural dependencies between the linguistic features and these language constructs. Guided by basic linguistic knowledge, we will detect sets of inter-dependent linguistic features and group them to a language construct consistent with the linguistic theory.

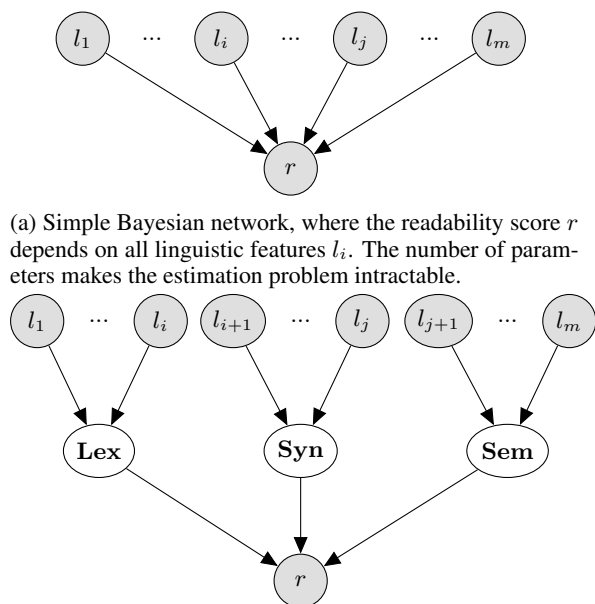
Examples of language constructs are the lexical difficulty **Lex**, the syntactic difficulty **Syn**, or the semantic difficulty **Sem**. Those variables cannot be directly measured in a text (because they are not well defined), but are rather unknown functions of some other linguistic features, such as the length of a word (in characters or syllables), the amount of uppercase letters (e.g. in acronyms) or the presence of digits (e.g. protein names in biology). The graphical representation that introduces language constructs as hidden variables can be found in Figure 1b. Those hidden variables are typically introduced in joint probabilistic models as:

$$\begin{aligned} \operatorname{Pr}(r, \mathcal{L}) &= \operatorname{Pr}(r, l_1, \dots, l_m) \\ &= \sum_{\mathbf{Lex}} \sum_{\mathbf{Syn}} \sum_{\mathbf{Sem}} \operatorname{Pr}(r, \mathbf{Lex}, \mathbf{Syn}, \mathbf{Sem}, l_1, \dots, l_m) \end{aligned} \quad (4)$$

By inspecting Figure 1b, we can observe that readability score  $r$  is independent from observable linguistic features  $l_i$  given the language constructs **Lex**, **Syn** and **Sem**. Thus, we can rewrite Equation 4 to factorize over the graph in Figure 1b as:

$$\begin{aligned} \operatorname{Pr}(r, l_1, \dots, l_m) &\approx \sum_{\mathbf{Lex}} \sum_{\mathbf{Syn}} \sum_{\mathbf{Sem}} p(r \mid \mathbf{Lex}, \mathbf{Syn}, \mathbf{Sem}) \\ &\quad \cdot p(\mathbf{Lex} \mid l_1, \dots, l_i) \cdot p(\mathbf{Syn} \mid l_{i+1}, \dots, l_j) \\ &\quad \cdot p(\mathbf{Sem} \mid l_{j+1}, \dots, l_m) \cdot p(l_1) \cdots p(l_m) \end{aligned} \quad (5)$$

where  $l_1, \dots, l_i$  are inter-dependent lexical features that somehow influence the lexical difficulty,



(a) Simple Bayesian network, where the readability score  $r$  depends on all linguistic features  $l_i$ . The number of parameters makes the estimation problem intractable.

(b) Structured Bayesian network that introduces language constructs (**Lex**, **Syn** and **Sem**) as hidden variables (white ellipses), with the purpose of reducing the dependencies of the readability score  $r$  from the rest of the linguistic features.

Figure 1: Graphical representations of causal networks. Arrows denote probabilistic influence.

$l_{i+1}, \dots, l_j$  are syntactic features that influence syntactic difficulty, and the remaining are semantic features. Now the readability score  $r$  depends only on a small set of language constructs, which dramatically reduces the amount of parameters.

### 3.3 Estimating Parameter Values

Hidden variables and independency assumptions are often necessary to reduce the number of parameters that need to be estimated, specially when there are many variables or there is a limited amount of training data. In the factor graph of Figure 2, there is a conditional probability distribution (CPD)  $\Pr(v \mid Pa_v)$  modeling the probability of every linguistic feature  $v$  given its parents  $Pa_v$  in the graph<sup>2</sup>. In this work, we make no assumptions on how a variable is related to its parents and we model this unknown relationship using non-parametric CPDs. The drawback is that we need to discretize the values of the linguistic variables and that the number of parameters<sup>3</sup> increases exponentially with the number of parent variables.

The estimation of the parameter values can be carried out using standard techniques that aim at optimizing the likelihood over the training data in presence of hidden variables. In this work,

<sup>2</sup>If a variable  $v$  has no parents, then its CPD is  $p(v)$ .

<sup>3</sup>The term “non-parametric” might be misleading, since this type of CPDs have many parameters.

we used the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) for that purpose.

### 3.4 Querying the Model

Estimating the joint probability distribution  $\Pr(r, \ell)$  has its advantages, since it gives us complete knowledge about the problem. In order to interpret the model, we can perform some insightful queries involving any variable.

**Marginal Maximum a Posteriori** is used to find the most probable value assignment to some linguistic features given some evidence. This query can mimic the functionality of discriminative models, where the objective is to find the most probable readability score  $\hat{r}$  given the linguistic evidence  $\ell^0$ , in presence of language constructs  $\mathbf{L}$ :

$$\text{MAP}(\hat{r} \mid \ell^0) = \underset{r}{\text{argmax}} \sum_{\mathbf{L}} p(r, \mathbf{L} \mid \ell^0) \quad (6)$$

where the conditional probability distribution  $p(r, \mathbf{L} \mid \ell^0)$  can be found by using the Bayes rule:

$$p(r, \mathbf{L} \mid \ell^0) = \frac{p(r, \mathbf{L}, \ell^0)}{p(\ell^0)} \quad (7)$$

Another application of marginal MAP queries is to gain linguistic insight about what characterizes unreadable texts. This insight could be obtained by querying the model in the opposite direction, i.e.  $\text{MAP}(\hat{\ell} \mid r^0)$ , where we want to obtain the most plausible linguistic instantiation  $\hat{\ell}$  given a certain readability level  $r^0$ . More complex queries can be similarly performed by conditioning the marginal MAP. For instance, the query  $\text{MAP}(\hat{\ell} \mid \text{Lex}^{\text{high}}, r^{\text{good}})$  would result in the most plausible values of linguistic features that have a high lexical difficulty but a good readability.

**Sensitivity analysis** allows us to understand how sensitive a certain variable is to some observed linguistic features. In our study, we are interested in understanding what individual or combination of observable linguistic features influence most in the readability of a particular text. A common approach (Kjærulff and Madsen, 2007) is to compute the distance  $d$  between the joint probability distribution with different instantiations of the linguistic features under study.

## 4 Experiments

We first describe the data that we used to train our systems, and the data grounded on cognitive

effort that we used for validation. Then, we describe the full set of linguistic features and our baseline systems. Finally, we assess to what extent our Bayesian causal network is able to predict the specific parts of the documents that are difficult to read, and compare it to other systems.

#### 4.1 Corpora

To estimate the parameters of the Bayesian causal network and our baseline systems, we opted to use texts from three corpora, namely Wikipedia Simple<sup>4</sup>, Wikipedia English<sup>5</sup>, and PubMed<sup>6</sup>.

Wikipedia has been a valuable resource for the development of text transformation methods, such as summarization (Biadys et al., 2008), or machine translation (Smith et al., 2010), among others. Wikipedia Simple is a relatively new version of the Wikipedia English, where articles are written in simple English<sup>7</sup>. Wikipedia English does not require any specific writing style other than clarity, precision and completeness. Finally, PubMed corpus is a large collection of academic biomedical articles, where readability is often sacrificed for precision and completeness. We assume that these three corpora have different expected readabilities (high, intermediate and low, respectively), and we use them as readability annotations at document level. Some linguistic features considered in our work are sensitive to text length (i.e. number of active lexical chains or average coreference distance). For this reason, we collected only abstracts from Wikipedia Simple that contain 10, 11 or 12 sentences, and randomly sampled from Wikipedia English and PubMed the same amount of long abstracts with the same text length distribution as Wikipedia Simple, totaling in 8, 856 abstracts.

Our hypothesis is that Bayesian causal networks are capable of recognizing specific parts of documents that make texts difficult to read. To test our hypothesis, we need documents with readability annotations at sub-document level. But such fine-grained annotations are difficult to obtain even for expert linguists because there are many linguistic variables involved in the annotation decisions.

In this work, we indirectly annotate the reading difficulty of every part of the text using an estimation of the expected cognitive effort required

to understand that part of the text. There are several methods that have been proposed to measure moment-to-moment cognitive effort, such as functional magnetic resonance imaging (fMRI) to quantify activations of certain brain areas, or measurements in pupil size changes. However, those methods have difficulties in aligning cognitive effort spatially and temporally to segments in a text, and we opted to measure fixation time on individual words due to its relative simplicity. Thus, we work under the assumption that higher cognitive effort is reflected as longer fixation durations, since parts of the text that are difficult to read require longer cognitive processing time.

On the text side, we characterize a part of a text by a quantification of its linguistic features at word level. Let  $f_{i,j}$  be the quantification of linguistic feature  $i$  at word  $w_j$ . As an example, linguistic feature “is noun”,  $f_{\text{noun},j} = 1$  if  $w_j$  is a noun. Non-binary linguistic features can be similarly quantified in the range  $[0, 1]$  dividing their value by the maximum possible value. For features not defined at word level (e.g. “sentence length”), the feature quantification of words in the span are all equal to the quantification of the span.

In order to estimate fixation time  $T_i$  induced by every linguistic feature  $i$ , we accumulate fixation durations on words scaled by the quantification of every linguistic feature at those words, and normalize it by the total amount of fixation durations and total amount of feature quantification. Formally, let  $t_j$  be the total amount of fixation duration on word  $w_j$ . Then, fixation time  $T_i$  caused by linguistic feature  $i$  can be computed as:

$$T_i = \frac{\sum_j t_j \cdot f_{i,j}}{(\sum_j t_j) \cdot (\sum_j f_{i,j})} \quad (8)$$

We collected fixation durations on every word using the eye-tracker Tobii TX300, and used a text-gaze aligner (Martínez-Gómez et al., 2012) to correct the systematic errors introduced by the eye-tracker. There were 40 subjects participating in our study, and only the 20% of eye-tracking sessions with highest signal quality were selected for this study. Most subjects were non-native English speakers linked to academia, with varying language skills and background knowledge. They were asked to carefully read 2 documents on 3 topics (6 documents in total), about economics, nutrition and astronomy, and answer detailed questionnaires to assess their understanding. The average

<sup>4</sup><http://simple.wikipedia.org/>

<sup>5</sup><http://en.wikipedia.org/>

<sup>6</sup><http://www.ncbi.nlm.nih.gov/pubmed/>

<sup>7</sup>Guidelines to write in simple English are proposed in Wikipedia Simple, but are not strictly enforced.



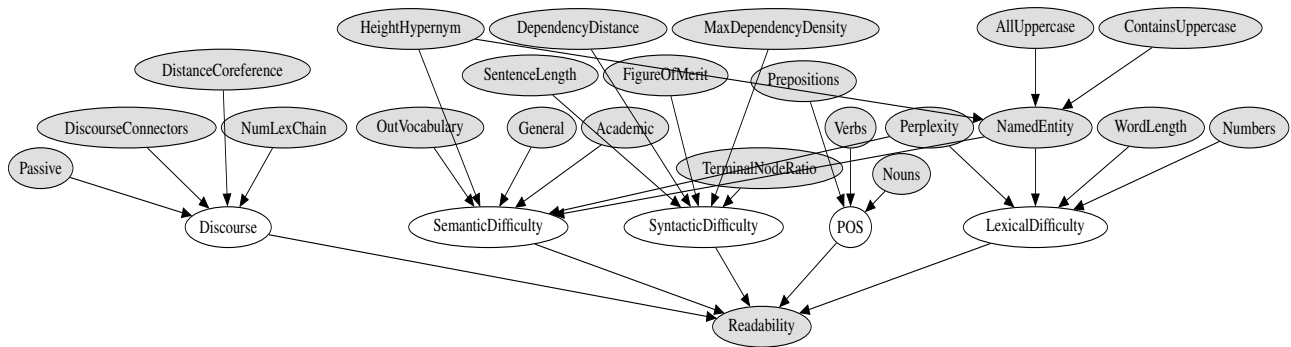


Figure 2: Graphical representation of our Bayesian causal network. Observable linguistic features are represented by white ellipses. Language constructs introduced as hidden variables are represented by gray ellipses. Directed edges indicate the direction of causality, and encode probabilistic influence.

duration of the reading and question-answering session was 1 hour, and every subject was compensated with the equivalent to 20 US dollars in cash at the end of the session. Documents contained 22.5 sentences and 469 words on average.

The objective of the Bayesian causal network will be to predict cognitive effort caused by each linguistic feature, and it will be compared to the results obtained by using discriminative methods.

#### 4.2 Feature Set

Figure 2 shows the 22 linguistic features (gray ellipses) that were used in this work, the 5 language constructs that were introduced as hidden variables (white ellipses), and their probabilistic relationships (directed edges). The linguistic features that appear as ancestors of lexical difficulty and Part of Speech (POS) correspond to their average at token level (i.e. in the case of “Numbers”, the percentage of tokens that are numbers).

Named entities were extracted using the NLTK toolkit (Bird et al., 2009), word lengths (in syllables) were computed by averaging the number of stresses in the CMU pronunciation dictionary (Weide, 1998). The perplexity was computed using Google 5-grams (Brants and Franz, 2006) with deleted interpolation tuned on a tokenized and non-lowercased separate subset of representative sentences from all three corpora. The percentage of prepositions, nouns and verbs was computed using the NLTK POS tagger.

Following the work in (Hudson, 1995) we considered the maximum dependency density and average distance between dependents as linguistic features that influence syntactic difficulty, computed using a dependency parser (Klein and Manning, 2003). Terminal node to non-terminal node

ratio is another typical phrase-based measure of syntactic difficulty, and it was computed using an HPSG parser (Miyao and Tsujii, 2008). The figure of merit, as given by the same parser, is a function of the lexical probability rules that are triggered during the automatic parsing, and somehow represents the parsing surprise.

Height of hypernyms were computed as the average distance between token lemmas to the most abstract term in WordNet (Fellbaum, 2010) and measures how specific terms are. “General”, “Academic” and “OutVocabulary” features denote the average number of words appearing in the General Word Service List (West and Jeffery, 1953), in the Academic Word List (Coxhead, 1998), or in none of them.

The average distance between mentions and their referents, and the maximum number of active lexical chains were computed using a coreference resolution system (Raghunathan et al., 2010) in a similar fashion to how the average dependency distance and maximum dependency density were computed to measure syntactic difficulty. Finally, the average number of passive clauses was computed using the output of the HPSG parser, and the percentage of tokens that are discourse connectors was measured checking the occurrence of every token in a hand-crafted list of 279 connectors.

#### 4.3 Baseline

Using our Bayesian network, we computed the importance of each linguistic feature for every document as the sensitivity of the network conditioned on the observation of the rest of the variables. We compared our system to two baselines. The first baseline, *raw features*, measures the importance of linguistic features (across the cor-

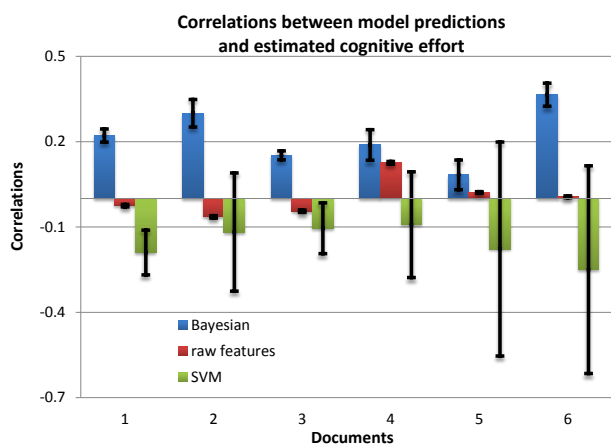


Figure 3: Correlations between predictions of feature impact on reading difficulty and the expected cognitive effort introduced by such features. Confidence intervals are computed at 95%.

pora) as the correlation between each linguistic feature and the readability score. As a second baseline, we chose SVM models due to their success in readability studies (François and Fairon, 2012). We measured their sensitivity to each linguistic feature, by observing the variations on the SVM response due to variations in each linguistic feature (Cortez, 2010), while holding the rest of linguistic variables to their average values. This baseline was built by training and tuning an SVM with a gaussian kernel in a cross-validation setup. Both baselines obtain quantifications of feature influence on readability independent of the document instantiation. For SVM sensitivity analysis, we also computed variability of SVM categorical response to changes in the linguistic feature under study while setting the rest of linguistic features to the instantiations on each document. However, there was a negligible variance in SVM response, and results are not reported for that experiment.

Confidence intervals for all systems were obtained by measuring prediction variability in 10 runs of random sampling with 90% of the data. All linguistic features were discretized in two intervals for the Bayesian network, except the readability score, which had three states (one for each class). This is an important loss of information for the Bayesian network, but it was necessary for computational reasons. SVM and raw features baselines, however, used continuous values.

#### 4.4 Results

Figure 3 shows correlations between predictions of feature impact on reading difficulty and the

expected cognitive effort introduced by such features. The  $x$ -axis corresponds to the identifier of each document for which we have estimated cognitive effort using the eye-tracker and the  $y$ -axis corresponds to correlations with the systems. Intervals for every prediction at a 95% confidence are displayed above and below each bar.

As it can be observed, raw features do not capture meaningfully cognitive effort and their correlations are close to zero, with a high confidence (narrow confidence intervals). The quantification on linguistic feature importance given by the SVM sensitivity analysis is slightly negative with large confidence intervals, which suggests that this type of analysis is not useful to predict reading difficulties in specific parts of the documents. The Bayesian causal network obtains mild, but consistent and positive correlations with the expected cognitive effort and its confidence intervals show strong significance.

Table 1 shows the most influential linguistic features on reading difficulty for documents 4 and 6. According to the cognitively-grounded reading difficulty, lexical perplexity (surprise), the occurrence of named entities, out of vocabulary words, passive clauses, academic words, nouns and abstraction (hypernyms) are the linguistic features that required longer fixation times in order to understand those documents. The Bayesian network ranked, on top 5, two and three of the most influential linguistic features for document 4 and 6.

## 5 Applications and Future Work

Bayesian causal networks for readability diagnosis have an immediate application to authoring systems, where the inference engine automatically detects text segments that make the text difficult to read. For that purpose, the average quantification of every linguistic feature has to be computed at document level. Then, causal reasoning (Bayesian sensitivity analysis) would be performed to find linguistic features with highest impact on reading difficulty for that specific document. Finally, instantiations of such linguistic features at segment level whose quantifications are above document average would be flagged for edition. Authors can then proceed to amend the text, or assert constraints. These constraints can take the form of “I want to increase readability without sacrificing the current lexical difficulty”. Such constraints can be introduced using marginal MAPs as described

Document 4	Cognitive effort	Bayesian	SVM	raw features
feature 1	<b>Nouns</b>	General	Dependency density	General
feature 2	<b>Out Vocabulary</b>	<b>Out Vocabulary</b>	General	<b>Out Vocabulary</b>
feature 3	<b>Passive</b>	<b>Academic</b>	Contains uppercase	<b>Academic</b>
feature 4	<b>Academic</b>	Figure of Merit	Dependency distance	All uppercase
feature 5	<b>Height Hypernym</b>	Named entities	Verbs	Figure of Merit
Document 6	Cognitive effort	Bayesian	SVM	raw features
feature 1	<b>Perplexity</b>	<b>Named entities</b>	Dependency density	General
feature 2	<b>Named entities</b>	<b>Academic</b>	General	<b>Out Vocabulary</b>
feature 3	<b>Out Vocabulary</b>	General	Contains uppercase	<b>Academic</b>
feature 4	<b>Passive</b>	<b>Perplexity</b>	Dependency distance	All uppercase
feature 5	<b>Academic</b>	Figure of Merit	Verbs	Figure of Merit

Table 1: List of 5 most influential linguistic features for documents 4 and 6, sorted in descending order. The first column corresponds to the order given by cognitive effort. The rest of the columns correspond to predictions of systems. The Bayesian network finds 2 and 3 out of the 5 most influential features in documents 4 and 6. SVM and raw features provide constant estimations for all documents.

in Section 3.4. There are, however, features that cannot be tweaked individually and would require very complex user actions. Others are simply very difficult to handle by humans, as in the case of the terminal node to non-terminal node ratio.

In an automatic readability optimization setup, a set of transformation actions could be applied on a text, but discerning the most appropriate action can be challenging. Bayesian networks could be a solution to it, since they can infer the desirable configuration of linguistic values for a certain readability level in a given document, and what actions would lead to the largest readability gain.

The remaining challenges when working with non-parametric Bayesian networks are two. The first one is the necessary loss of information that occurs when discretizing features, and parametric models are possible solutions. Finding better network topologies is also an interesting challenge that brings linguistic insights into readability studies and increases the predictive power of the model. One approach is to refine the network using more thoughtful linguistic knowledge. Another possibility is to automatically estimate the optimal network topology driven by data, but causal properties could be difficult to preserve.

We used indirect measurements of cognitive effort that rely on the computation of a normalized fixation time on every linguistic feature. Fixation durations were recorded using a precise eye-tracker, but data collection is rarely exempt of systematic errors and new methods to estimate cognitive effort should account for this degraded calibration. Moreover, certain aspects of cognitive ef-

fort might not be reflected by fixation times, and other features of eye movements, such as regressions or changes in pupil diameter can be valuable.

Since estimations of feature impact on readability depends on each document, it was difficult to compare our findings to prior work. Future investigations in readability diagnosis would benefit from a combination of indirect measurements of cognitive effort and readability annotations by linguistic experts at sub-document level, that could be shared within the research community.

## 6 Conclusions

Discriminative models are built to predict readability and correlate well with human judgment. Those models are good readability predictors, but fail at explaining the causes of unreadability. With the intention of assisting humans to optimize readability or to fully automate it, we need methods able to infer the causes of readability.

We have presented the application of Bayesian causal networks to build generative readability models. To reduce the number of dependencies between linguistic features, we introduced language constructs as hidden variables and estimated the parameter values using the EM algorithm.

Using our proposed Bayesian causal network, we measured the impact of every linguistic feature in presence of all other variables, and compared the prediction accuracy to grounded cognitive effort. Our method showed significant and positive correlations with cognitive effort, suggesting that it is able to capture linguistic features that cause difficulties in reading for specific documents.

## References

- S. Aluísio, L. Specia, C. Gasperin, and C. Scarton. 2010. Readability assessment for text simplification. In *Proc. of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9. ACL.
- F. Biadys, J. Hirschberg, and E. Filatova. 2008. An unsupervised approach to biography production using Wikipedia. In *Proc. of ACL-08: HLT*, pages 807–815.
- S. Bird, E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. O’Reilly Media Inc.
- T. Brants and A. Franz. 2006. Web 1T 5-gram vers. 1.
- J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. Tait. 1999. Simplifying text for language-impaired readers. In *Proc. of EACL*, volume 99, pages 269–270.
- P. Cortez. 2010. Data mining with neural networks and support vector machines using the r/miner tool. In *Advances in Data Mining. Applications and Theoretical Aspects*, pages 572–583. Springer.
- A. Coxhead. 1998. *An academic word list*, volume 18. School of Linguistics and Applied Language Studies, Victoria University of Wellington.
- A. Davison and R.N. Kantor. 1982. On the failure of readability formulas to define readable texts: A case study from adaptations. *Reading Research Quarterly*, pages 187–209.
- A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*.
- S. Devlin, J. Tail, Y. Canning, J. Carroll, G. Minnen, and D. Pearce. 1999. The application of assistive technology in facilitating the comprehension of newspaper text by aphasic people. *Assistive Technology on the Threshold of the New Millennium*.
- C. Fellbaum. 2010. WordNet. *Theory and Applications of Ontology: Computer Applications*, pages 231–243.
- R. Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.
- T. François and C. Fairon. 2012. An AI readability formula for French as a foreign language. In *Proc. of the 2012 Joint Conference on EMNLP and CoNLL*, pages 466–477. ACL.
- E. Fry. 1990. A readability formula for short passages. *Journal of Reading*, pages 594–597.
- M. J. Heilman, K. Collins-Thompson, J. Callan, and M. Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Proc. of NAACL HLT*, pages 460–467.
- R.A. Hudson. 1995. Measuring syntactic difficulty. *Manuscript, University College, London*.
- R.J. Kate, X. Luo, S. Patwardhan, M. Franz, R. Florian, R.J. Mooney, S. Roukos, and C. Welty. 2010. Learning to predict readability using diverse linguistic features. In *Proc. of the 23rd COLING*, pages 546–554. ACL.
- J.P. Kincaid, R. Fishburne, R.L. Rogers, and B.S. Chissom. 1975. Derivation of new readability formulas for navy enlisted personnel.
- U. B. Kjærulff and A. L. Madsen. 2007. *Bayesian Networks and Influence Diagrams*. Information science and statistics. Springer New York.
- D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of the 41st ACL*, pages 423–430. ACL.
- P. Martínez-Gómez, C. Chen, T. Hara, Y. Kano, and A. Aizawa. 2012. Image registration for text-gaze alignment. In *Proc. of the IUI’12*, pages 257–260.
- G.H. Mc Laughlin. 1969. SMOG grading—a new readability formula. *Journal of reading*, pages 639–646.
- Y. Miyao and J. Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34:35–80, March.
- S.E. Petersen and M. Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer Speech & Language*, 23(1):89–106.
- K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. 2010. A multi-pass sieve for coreference resolution. In *Proc. of the 2010 on EMNLP*, pages 492–501. ACL.
- K. Rayner and S.A. Duffy. 1986. Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14(3):191–201.
- L. Si and J. Callan. 2001. A statistical model for scientific readability. In *Proc. of the 10th CIKM*, pages 574–576. ACL.
- A. Siddharthan. 2003. *Syntactic Simplification and Text Cohesion*. Ph.D. thesis, University of Cambridge.
- J.R. Smith, C. Quirk, and K. Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *HLT: The 2010 NA-ACL*, pages 403–411. ACL.
- R.L. Weide. 1998. The CMU pronunciation dictionary, release 0.6.
- M. West and G.B. Jeffery. 1953. *A general service list of English words*. Longmans, Green London.
- S. Williams and E. Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proc. of the 10th European Workshop on Natural Language Generation*.

# Word Co-occurrence Counts Prediction for Bilingual Terminology Extraction from Comparable Corpora

Amir Hazem and Emmanuel Morin

Laboratoire d'Informatique de Nantes-Atlantique (LINA)

Université de Nantes, 44322 Nantes Cedex 3, France

{Amir.Hazem, Emmanuel.Morin}@univ-nantes.fr

## Abstract

Methods dealing with bilingual lexicon extraction from comparable corpora are often based on word co-occurrence observation and are by essence more effective when using large corpora. In most cases, specialized comparable corpora are of small size, and this particularity has a direct impact on bilingual terminology extraction results. In order to overcome insufficient data coverage and to make word co-occurrence statistics more reliable, we propose building a predictive model of word co-occurrence counts. We compare different predicting models with the traditional *Standard Approach* (Fung, 1998) and show that once we have identified the best procedures, our method increases significantly the performance of extracting word translations from comparable corpora.

## 1 Introduction

Using comparable corpora for bilingual lexicon extraction is becoming more and more a matter of interest, especially because of the easier availability of this kind of corpora comparing to parallel ones. Many researchers proposed a variety of approaches (Fung, 1995; Rapp, 1999; Chiao and Zweigenbaum, 2002; Déjean et al., 2002; Morin et al., 2007; Laroche and Langlais, 2010, among others). While different improvements were achieved, the starting point remains words'co-occurrences as they represent the observable evidence that can be distilled from a corpus. Hence, frequency counts for word pairs often serve as a basis for distributional methods. The main assumption underlying bilingual lexicon extraction is: two words are more likely to be a translation of each other if they share the same lexi-

cal contexts (Fung, 1998). The most popular approach named, the *Standard Approach* (Fung and Mckeown, 1997; Rapp, 1999), makes use of this assumption to perform bilingual lexicon extraction. While good results on single word terms (SWTs) can be obtained from large corpora of several million words (80% for the top 10-20 (Fung and Mckeown, 1997), 91% accuracy for the top 3 (Cao and Li, 2002)). Results drop significantly using specialized small corpora (60% for the top 20 (Chiao and Zweigenbaum, 2002; Déjean et al., 2002; Morin et al., 2007)).

The reliability of co-occurrence counts greatly relies on the amount of data. Clearly, the larger the training corpus, the more representative it is likely to be, and thus the more reliable the statistics of words. Therefore, the number and distribution of types in the available small sample are not reliable estimators (Evert and Baroni, 2007). This latter fact motivates the necessity of an alternative to the unreliable counts especially when using small specialized comparable corpora. Statistical NLP often deals in the prediction of variables ranging from text categories to linguistic structures to novel utterances. If large specialized comparable corpora are not available, one way to approach this problem and to make co-occurrence counts more reliable, is to use prediction models of word co-occurrence counts based on large training datasets. Corpus data from the general domain such as newspapers, for instance, is abundant and can be easily used for training.

The main contribution of this paper is to investigate different word co-occurrence prediction models for the task of bilingual terminology extraction from comparable corpora. Our aim is to make the observed word co-occurrence counts in small specialized comparable corpora more reliable by re-estimating their probabilities. For that purpose we explore different models such as the linear regression often used to model data using

linear predictor functions, the mean average word co-occurrence increase and the Good-Turing estimator. All of the predicting models rely on the observed counts of word co-occurrence in a training dataset of small and large corpora from the general domain. While prediction is widely used in NLP, to our knowledge no investigation of co-occurrence prediction for the task of bilingual terminology extraction from comparable corpora has been addressed so far. We show that using our method as a pre-processing step of the *Standard Approach*, leads to significant improvements on the performance of bilingual terminology extraction.

In the remainder of this paper, we present in section 2 the related work on bilingual lexicon extraction from comparable corpora. Then, we introduce in section 3 the *Standard Approach* used as baseline. Section 4 describes our method and the different predicting models of co-occurrence counts. Section 5 describes the different linguistic resources used in our experiments. Section 6 evaluates the contribution of the predicting models on the quality of bilingual terminology extraction through different experiments. We discuss our findings in section 7 and finally conclude in section 8.

## 2 Related Work

The distributional hypothesis which states that words with similar meaning tend to occur in similar contexts, has been extended to the bilingual scenario (Fung, 1998; Rapp, 1999). Hence, using comparable corpora, a translation of a source word can be found by identifying a target word with the most similar context. A popular method often used as a baseline is the *Standard Approach* (Fung, 1998). It consists of using the bag-of-words paradigm to represent words of source and target language by their context vector. After word contexts have been weighted using an association measure (the point-wise mutual information (Fano, 1961), the log-likelihood (Dunning, 1993), the discounted odds-ratio (Laroche and Langlais, 2010)), the similarity between a source word's context vector and all the context vectors in the target language is computed using a similarity measure (cosine (Salton and Lesk, 1968), Jaccard (Grefenstette, 1994)...). Finally, the translation candidates are ranked according to their similarity score.

Many variants of the *Standard Approach* have been proposed. They can differ in context representation (window-based, syntactic-based) (Morin et al., 2007; Gamallo, 2008), corpus characteristics (small, large, general or domain specific...)(Chiao and Zweigenbaum, 2002; Déjean et al., 2002; Morin et al., 2007), type of words to translate (single word terms (SWTs) or multi-word terms (MWTs))(Rapp, 1999; Daille and Morin, 2005), words frequency (less frequent, rare...)(Pekar et al., 2006), etc.

There exist other approaches for bilingual lexicon extraction. Déjean et al. (2002) introduce the *Extended Approach* to avoid the insufficient coverage of the bilingual dictionary required for the translation of source context vectors. A variation of the latter method based on centroid is proposed by Daille and Morin (2005). Haghghi et al. (2008) employ dimension reduction using canonical component analysis (CCA) and Rubino and Linares (2011) propose a multi-view approach based on linear discriminant analysis (LDA) among others.

## 3 Standard Approach

The *Standard Approach* is based on words co-occurrence vectors. The basic idea is to go through a corpus and to count the number of times  $n(c, t)$  each context word  $c$  occurs within a window of a certain size  $w$  around each target word  $t$ . According to (Fung and Mckeown, 1997; Fung, 1998; Rapp, 1999), the *Standard Approach* can be carried out as follows:

For a source word to translate  $w_i^s$ , we first build its context vector  $v_{w_i^s}$ . The vector  $v_{w_i^s}$  contains all the words that co-occur with  $w_i^s$  within windows of  $n$  words. Let's denote by  $coocc(w_i^s, w_j^s)$  the co-occurrence value of  $w_i^s$  and a given word of its context  $w_j^s$ . The process of building context vectors is repeated for all the words of the target language. An association measure such as the point-wise mutual information (Fano, 1961), the log-likelihood (Dunning, 1993) or the discounted odds-ratio (Laroche and Langlais, 2010) is used to score the strength of correlation between a word and all the words of its context vector. The context vector  $v_{w_i^s}$  is projected into the target language  $v_{w_i^t}$ . Each word  $w_j^s$  of  $v_{w_i^s}$  is translated with help of a bilingual dictionary  $D$ . If  $w_j^s$  is not present in  $D$ ,  $w_j^s$  is discarded. Whenever the bilingual dictionary provides several translations for a word, all the entries are considered but weighted according

to their frequency in the target language (Morin et al., 2007). A similarity measure is used to score each target word  $w_i^t$ , in the target language with respect to the translated context vector,  $v_{w_i^s}^t$ . Usual measures of vector similarity include the cosine similarity (Salton and Lesk, 1968) or the weighted Jaccard index (Grefenstette, 1994) for instance. The candidate translations of the word  $w_i^s$  are the target words ranked following the similarity score.

## 4 Method

### 4.1 Basic Idea

We start from the assumption that words that co-occur together more often than by chance in a small corpus, should have the same behaviour in a bigger corpus with higher co-occurrence values. Our aim is to estimate the increasing values. We choose to observe co-occurrence counts using a training dataset. Table 1 shows the increase of word co-occurrence counts in corpus of different sizes. Let's denote by  $w_i$  and  $w_j$  two given words. We take as a starting point a corpus of 500,000 words in English, French and Spanish then, for each couple of words  $(w_i, w_j)$  that occur together, we observe their co-occurrence count variation in corpus of 1, 2 and 5 million words per language. For instance, if  $coocc(w_i, w_j) = 5$  in the English corpus of 500,000 words, we observe  $coocc(w_i, w_j)$  in the English corpus of 1, 2 and 5 million words and observe how much this value increases.

Table 1 can be read as follows: Let's take  $coocc_{En} > 1$ , we can see in Table 1 that there is an increase of 37.19% of words that co-occur more than 1 time in the corpus of 1 million words, 57.06% in the corpus of 2 million words and 73.02% in the corpus of 5 million words. The observations of Table 1 confirm that word co-occurrence counts increase in most cases (97.34% for  $coocc_{Es} > 4$ , 98.87% for  $coocc_{Fr} > 4...$ )

### 4.2 Co-occurrence Counts Estimation

Let's denote by  $E_S = \{v_S^1, v_S^2, \dots, v_S^n\}$  the set of the observed co-occurrence counts in a small training corpus. Our aim is to estimate the expected co-occurrence counts  $E_L = \{v_L^1, v_L^2, \dots, v_L^n\}$  in a large corpus. To do so, one intuitive way for estimation is the mean average increase (MAI) of each co-occurrence count. A more effective model that has proven its efficiency is linear regression. For that reason, we decided to use lin-

#Co-occ	1m	2m	5m
$coocc_{En} > 0$	16.13	30.04	47.06
$coocc_{En} = 1$	10.99	23.39	40.66
$coocc_{En} > 1$	37.19	57.06	73.02
$coocc_{En} > 2$	57.50	77.50	88.54
$coocc_{En} > 3$	68.83	85.39	92.63
$coocc_{En} > 4$	77.41	90.79	95.65
$coocc_{En} > 5$	82.11	92.70	96.28
$coocc_{Fr} > 0$	17.74	30.50	47.76
$coocc_{Fr} = 1$	12.55	24.04	41.58
$coocc_{Fr} > 1$	47.84	67.79	83.39
$coocc_{Fr} > 2$	69.28	86.95	95.30
$coocc_{Fr} > 3$	80.81	93.68	98.00
$coocc_{Fr} > 4$	87.93	96.76	98.87
$coocc_{Fr} > 5$	91.30	97.84	99.14
$coocc_{Es} > 0$	18.64	35.50	51.27
$coocc_{Es} = 1$	13.15	28.55	44.91
$coocc_{Es} > 1$	40.99	63.60	76.92
$coocc_{Es} > 2$	60.93	82.93	91.42
$coocc_{Es} > 3$	71.60	89.40	94.80
$coocc_{Es} > 4$	78.91	93.74	97.03
$coocc_{Es} > 5$	83.13	95.06	97.34

Table 1: Word co-occurrence counts increase (%) in corpus of different sizes on the English, French and Spanish Newspapers

ear regression (*LReg*) for prediction. In statistical NLP, smoothing techniques for n-gram models have been addressed in a number of studies (Chen and Goodman, 1999). We chose to apply the simple Good-Turing estimator (Good, 1953) as it is an appropriate way to estimate word co-occurrence counts. We finally present a naive model based on the maximum (*Max*) and the mean average count (*Mean*) of observed word co-occurrence counts in a small and large training datasets.

#### 4.2.1 Mean Average Increase

Results shown in Table 1 lead to an intuitive model which consists of the estimation of the mean average increase of each co-occurrence count. To estimate  $E_L$  we use a training corpus divided in two sets of small (500,000 words) and large (10 million words) corpus. Hence, we estimate the increasing value for each co-occurrence pair count. Let's denote by:

$E_S^1 = \{coocc_S^1(w_i, w_j) = 1, i \in [1, N], j \in [1, M]\}$   
the set of co-occurrence pairs of count 1 observed in a small corpus and by:

$E_L^o = \{coocc_L^1(w_i, w_j) = o_{ij}, i \in [1, N], j \in [1, M]\}$   
the set of co-occurrence pairs of count  $o_{ij}$  observed in a large corpus. The mean average increase  $MAI_1$  for 1 count co-occurrence pairs is:

$$MAI_1 = \frac{1}{|E_S^1|} \sum_{i=1}^N \sum_{j=1}^M (\text{coocc}_L^1(w_i, w_j) - \text{coocc}_S^1(w_i, w_j)) \quad (1)$$

The generalized formula for a given pair co-occurrence count  $k$  is:

$$MAI_k = \frac{1}{|E_S^k|} \sum_{i=1}^N \sum_{j=1}^M (\text{coocc}_L^k(w_i, w_j) - \text{coocc}_S^k(w_i, w_j)) \quad (2)$$

#### 4.2.2 Good-Turing Estimator

Smoothing techniques (Good, 1953) are often used to better estimate probabilities when there is insufficient data to estimate probabilities accurately. They tend to make distributions more uniform, by adjusting low probabilities such as zero probabilities upward, and high probabilities downward. The Good-Turing estimator (Good, 1953) states that for any  $n$ -gram that occurs  $r$  times, we should pretend that it occurs  $r^*$  times. The Good-Turing estimator uses the count of things you have seen once to help estimate the count of things you have never seen. In order to compute the frequency of words, we need to compute  $N_c$ , the number of events that occur  $c$  times (assumes that all items are binomially distributed). Let  $N_r$  be the number of items that occur  $r$  times.  $N_r$  can be used to provide a better estimate of  $r$ , given the binomial distribution. The adjusted frequency  $r^*$  is then:

$$r^* = (r + 1) \frac{N_{r+1}}{N_r} \quad (3)$$

The function  $r^*$  is applied to all the observed co-occurrence counts of the test data.

#### 4.2.3 Linear Regression

Starting from the observations in Table 1, thanks to linear regression we attempt to model the relationship between the first variable which corresponds to the co-occurrence distribution of words in the small corpus known as the explanatory variable, and the second variable which corresponds to the co-occurrence distribution of words in the large corpus known as the dependent variable. Before applying the linear regression we want to ensure that there is a correlation between the two variables; to do so, we apply the correlation coefficient as presented in Table 2:

Cor	1m	2m	5m
$cor_{En}$	0.933	0.894	0.788
$cor_{Fr}$	0.924	0.899	0.872
$cor_{Es}$	0.904	0.854	0.801

Table 2: Word co-occurrence counts correlation between corpus of 500,000 words and corpus of different sizes (1 million, 2 million and 5 million words) on the English, French and Spanish Newspaper

We can see according to Table 2 that there is a strong correlation of word co-occurrence counts across corpora of different sizes. Let's denote by  $f$  the linear function of explanatory variables. We use in our case one explanatory variable  $X$  that corresponds to the set of word co-occurrence counts in a small corpus.

- $Y = \beta_1 X + \beta_0$
- For each  $x$  of  $X$ :  $f(x) = \beta_1 x + \beta_0$

By applying linear regression to our training dataset we obtain the following equations:

For the English corpus we obtain:

$$Y_{1m} = 1.742X - 0.686$$

$$Y_{2m} = 3.184X - 2.008$$

$$Y_{5m} = 5.997X - 3.967$$

For the French corpus we obtain:

$$Y_{1m} = 1.802X - 0.673$$

$$Y_{2m} = 3.104X - 1.773$$

$$Y_{5m} = 7.167X - 5.137$$

Where  $Y_{1m}$  for instance, corresponds to the linear regression function learned from the training corpus of 1 million words.

#### 4.2.4 Mean and Max Models

As shown in Table 1, co-occurrence counts increase automatically when corpus size increases. A straightforward and maybe naive process is to select the observed counts of co-occurrence pairs in the training large corpus as the new estimation



values. Hence, using the mean process, each co-occurrence pair count can be estimated as follows:

$$\text{Mean}_k = \frac{1}{N} \sum_{i=1}^N \text{count}(k, i) \quad (4)$$

Where  $k$  is the observed count in the small corpus and  $i$  is the observed count in the large corpus of a given words pair. In the same way, using the max process, each co-occurrence pair count is estimated as follows:

$$\text{Max}_k = \frac{1}{N} \text{MAX}_{i=1}^N \text{count}(k, i) \quad (5)$$

## 5 Linguistic Resources

In order to evaluate the prediction techniques, several linguistic resources are needed. We present hereafter the comparable corpora, the bilingual dictionary and the reference lists used in our experiments.

### 5.1 Corpus Data

Experiments have been carried out on two English-French comparable corpora. A specialized corpus of 1 million words from the medical domain within the sub-domain of breast cancer and a specialized corpus from the domain of wind energy of 600,000 words.

	Breast cancer	Wind energy
$Tokens_S$	500,000	300,000
$Tokens_T$	500,000	300,000
$ S $	8,221	6,081
$ T $	6,631	5,606

Table 3: Corpus size

For the breast cancer corpus, we have selected the documents from the Elsevier website<sup>1</sup> in order to obtain an English-French specialized comparable corpora. We have automatically selected the documents published between 2001 and 2008 where the title or the keywords contain the term 'cancer du sein' in French and 'breast cancer' in English. For the wind energy corpus, we used the *Babook* crawler (Groc, 2011) to collect documents in French and English from the web. As the documents were collected from different websites according to some keywords of the domain,

<sup>1</sup>[www.elsevier.com](http://www.elsevier.com)

this corpus is more noisy and less well structured comparing to the breast cancer corpus. The two bilingual corpora have been normalized through the following linguistic pre-processing steps: tokenization, part-of-speech tagging, and lemmatization. The function words have been removed and the words occurring once (i.e. hapax) in the French and the English parts have been discarded. As summarized in Table 3, The breast cancer corpus comprised about 8,221 distinct words in English ( $|S|$ ) and 6,631 distinct words in French ( $|T|$ ). The wind energy corpus comprised about 6,081 distinct words in English ( $|S|$ ) and 5,606 distinct words in French ( $|T|$ ).

### 5.2 Dictionary

We used in our experiments the French-English bilingual dictionary ELRA-M0033 of about 200,000 entries<sup>2</sup>. It contains, after linguistic pre-processing steps and projection on both corpora less than 4000 distinct words. The details are given in Table 4.

	Breast cancer	Wind energy
$ ELRA_S $	3,573	3,459
$ ELRA_T $	3,670	3,326

Table 4: Dictionary coverage

### 5.3 Reference Lists

To build our reference lists, we selected only the English/French pair of single-word terms (SWTs) which occur more than five times in each part of the comparable corpus. As a result of filtering, 321 English/French SWTs were extracted (from the UMLS<sup>3</sup> meta-thesaurus) for the breast cancer corpus and 100 pairs for the wind energy corpus. The small size of the reference lists can be explained by the fact that small specialized comparable corpora contain a limited set of specialized terms. We can also notice that in bilingual terminology extraction from specialized comparable corpora, the terminology reference list is often composed of 100 SWTs (180 SWTs in (Déjean et al., 2002), 95 SWTs in (Chiao and Zweigenbaum, 2002), and 100 SWTs in (Daille and Morin, 2005)).

<sup>2</sup>ELRA dictionary has been done by Sciper in the Technolange/Euradic project

<sup>3</sup><http://www.nlm.nih.gov/research/umls>

## 5.4 Training Dataset

Predicting models such as linear regression or the Good-Turing estimator need a large training corpus to estimate the adjusted co-occurrences. For that purpose, we chose a training corpus composed of two sets. A small set of 500,000 words and a large set of 10 million words. We selected the documents published in 1994 from newspapers *Los Angeles Times/Le Monde*.

## 6 Experiments and Results

The baseline in our experiments is the *Standard Approach* (Fung, 1998) which is often used for comparison (Pekar et al., 2006; Gamallo, 2008; Prochasson and Morin, 2009), etc. In this section, we first give the parameters of the *standard approach*, then we present the results of the experiments conducted on the two corpora presented above: 'Breast cancer' and 'Wind energy'.

### 6.1 Experimental Setup

Using the *Standard Approach*, three major parameters need to be set:

1. The size of the window used to build the context vectors (Morin et al., 2007; Gamallo, 2008)
2. The association measure (the log-likelihood (Dunning, 1993), the point-wise mutual information (Fano, 1961), the discounted odds-ratio (Laroche and Langlais, 2010)...) )
3. The similarity measure (the weighted Jaccard index (Grefenstette, 1994), the cosine similarity (Salton and Lesk, 1968),...)

Laroche and Langlais (2010) carried out a complete study of the influence of these parameters on the quality of bilingual lexicon extraction from comparable corpora. To build the context vectors we chose a 7-window size. The entries of the context vectors were determined by the log-likelihood, the point-wise mutual information and the discounted odds-ratio. As similarity measure, we chose to use the weighted Jaccard index and the cosine similarity. Other combinations of parameters were assessed but the previous parameters turned out to give the best performance.

We note that *Top k* means that the correct translation of a given word is present in the k first candidates of the list returned by the *Standard Approach*. We use also the mean average precision

*MAP* (Manning and Schutze, 2008) which represents the quality of the system.

$$MAP(Q) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{m_i} \sum_{m_i=1}^k P(R_{ik}) \quad (6)$$

where  $|Q|$  is the number of terms to be translated,  $m_i$  is the number of reference translations for the  $i^{th}$  term (always 1 in our case), and  $P(R_{ik})$  is 0 if the reference translation is not found for the  $i^{th}$  term or  $1/r$  if it is ( $r$  is the rank of the reference translation in the translation candidates).

### 6.2 Results

We conducted a set of two experiments on two specialized comparable corpora. We carried out a comparison between the *Standard Approach* (*SA*) and the different prediction models presented in section 4.2 namely: the maximum model (*Max*), the mean model (*Mean*), the linear regression model (*LReg*), The Good-Turing estimator (*GT*) and the mean average increase model (*MAI*). Experiment 1 shows the results on the breast cancer corpus and experiment 2 those of the wind energy corpus.

Table 5 shows the results of the experiments on the breast cancer corpus. The first observation concerns the *Standard Approach* (*SA*). The best results are obtained using the Log-Jac parameters with a MAP of 27.9%. We can also notice that for this configuration, none of the prediction models improve the performance of the *Standard Approach*. On the contrary, they even degrade the results. The second observation concerns the Odds-Cos parameters where the naive *Mean*, *Max* and *MAI* models are under the baseline. The best score is obtained by the *LReg* model with a MAP of 27.6%. The most notable result concerns the PMI-Cos parameters. We can notice that four of the five techniques improve the performance of the baseline. The best prediction model is the *Max* technique which reaches a MAP of 27.2% and improves the Top1 precision of 4.8% and the Top10 precision of 6.6%.

Table 6 shows the results of the experiments on the wind energy corpus. Generally the results follow the same behaviour as the previous experiment. The best results of the *Standard Approach* are obtained using the Log-Jac parameters with a MAP of 25.7%. Here also, none of the prediction models improve the performance of the *Standard Approach*. About the Odds-Cos parameters,

	SA	Max	Mean	LReg	MAI	GT	
P1	15.5	<b>20.2</b>	13.7	18.0	18.6	18.6	PMI-Cos
P5	31.1	<b>35.8</b>	28.3	<b>35.8</b>	34.2	32.0	
P10	34.5	41.1	32.7	<b>42.0</b>	38.3	37.0	
MAP	22.6	<b>27.2</b>	20.3	26.7	26.4	25.6	
P1	15.8	15.5	11.8	<b>19.9</b>	13.7	16.8	Odds-Cos
P5	<b>34.8</b>	30.2	28.6	34.2	27.7	34.2	
P10	40.4	36.7	35.5	<b>41.7</b>	33.0	39.8	
MAP	24.8	22.9	19.8	<b>27.6</b>	20.9	25.2	
P1	<b>20.2</b>	06.5	16.5	15.5	09.9	14.6	Log-Jac
P5	<b>35.8</b>	15.5	33.9	28.6	21.4	27.7	
P10	<b>42.6</b>	20.5	38.3	37.3	26.7	34.2	
MAP	<b>27.9</b>	11.6	24.6	22.6	15.6	21.4	

Table 5: Results of the experiments on the 'Breast cancer' corpus (the improvements indicate a significance at the 0.05 level using Student's t-test).

	SA	Max	Mean	LReg	MAI	GT	
P1	07.0	13.0	10.0	<b>18.0</b>	15.3	14.0	PMI-Cos
P5	27.0	34.0	30.0	<b>37.0</b>	33.0	31.0	
P10	37.0	<b>46.0</b>	36.0	<b>46.0</b>	43.0	43.0	
MAP	17.8	23.1	19.2	<b>28.0</b>	25.0	22.9	
P1	12.0	09.0	06.0	<b>14.0</b>	10.0	12.0	Odds-Cos
P5	31.0	20.0	27.0	<b>32.0</b>	25.0	31.0	
P10	38.0	26.0	39.0	<b>40.0</b>	33.0	36.0	
MAP	21.8	15.7	17.0	<b>23.3</b>	18.0	19.8	
P1	<b>17.0</b>	09.0	18.0	15.0	18.0	13.0	Log-Jac
P5	<b>36.0</b>	16.0	30.0	31.0	29.0	27.0	
P10	<b>42.0</b>	22.0	45.0	36.0	36.0	37.0	
MAP	<b>25.7</b>	14.0	25.1	22.9	23.7	20.5	

Table 6: Results of the experiments on the 'Wind energy' corpus (the improvements indicate a significance at the 0.05 level using Student's t-test).

here again the naive *Mean*, *Max* and *MAI* models are under the baseline. We can also notice that *GT* is slightly under the *standard approach*. The best score is obtained by the *LReg* model with a MAP of 23.3%. Finally, the most remarkable result still concerns the PMI-Cos parameters where the same four of the five predicting techniques improve the performance of the baseline. The best prediction model is the *LReg* technique which reaches a MAP of 28.0% and improves the Top1 precision of 11.0% and the Top10 precision of 10.2%.

## 7 Discussion

The aim of this work was to propose and contrast different word co-occurrence prediction approaches: naive or intuitive (*Max*, *Mean* and *MAI*) and more sophisticated (*LReg* and *GT*)

aiming to improve bilingual terminology extraction. Our approach can be used as a pre-processing step of the *Standard Approach* by applying a prediction function to word co-occurrence counts.

According to the experimental results, the first observation is that the *Standard Approach* performs better when using the log-likelihood measure comparatively to the discounted odds-ratio and the point-wise mutual information measures. This supposes that the log-likelihood provides a better estimation of word co-occurrence counts. The log-likelihood measures significance (i.e. the amount of evidence against the null hypothesis) and is known to be more robust against low expected frequencies (Dunning, 1993). The lower performance of the *Standard Approach* when using the point-wise mutual information is certainly due to the over-estimation of low frequencies. In practical applications, PMI was found to have a tendency to assign inflated scores to low-frequency word pairs. Thus, even a single co-occurrence of two words might result in a fairly high association score. The discounted odds-ratio has shown lower results when compared to log-likelihood unlike its better performance as shown in Laroche and Langlais (2010). This is certainly due to the multiple parameters and resources of the *Standard Approach* and also the cosine similarity measure which is sensitive to context vector size. In our experiments, we did not investigate this parameter as it is not the matter of our study. We considered the whole context vector of each word.

According to the PMI-Cos configuration, the baseline is consistently outperformed by every prediction model (except *Mean* on the breast cancer experiment). The good results of the proposed methods when associated to the PMI-Cos configuration suggest that the over-estimation of infrequent counts of PMI is skimmed by the prediction function. This finding can be considered as a new way to counterbalance the low-frequency bias of PMI. The best prediction approach shown in the experiments is *Max* with a MAP of 27.2%, followed by *LReg* with a MAP of 26.7% on the breast cancer corpus. Nevertheless, in the wind energy corpus *LReg* performed substantially better than *Max* with a MAP of 28.0% while *Max* reaches 23.1% only. Even the lower performance of *MAI* and *GT*, they also provide significant improvements.

In our experiments, none of the proposed algo-

rithms reached good results while associated to the Log-Jac configuration. This is certainly related to the properties of the log-likelihood association measure. While the prediction models tend to increase small co-occurrence counts, this can lead to the overrating of infrequent words renders the ranking of the log-likelihood measure useless. Concerning the Odds-Cos parameters, although there were slight improvements on the *LReg* algorithm, other methods have shown disappointing results. Here again the Odds-ratio association measure seems to be not compatible with re-estimating co-occurrence counts. More investigations are certainly needed to highlight the reasons of this poor performance. It seems that prediction functions do not fit well with association measures based on contingency table.

The most noticeable improvement concerns the PMI-Cos configuration. Aside from the *Mean* method, all the other techniques have shown better performance than the *Standard Approach*. According to the empirical results, point-wise mutual information performs better with *Max* and *LReg* techniques. Furthermore and as has been pointed out above, prediction models seem to be an alternative to the low-frequency bias of the point-wise mutual information. It is our hope that the present work may provide a starting point to co-occurrence prediction on comparable corpora as an alternative to unreliable counts. The next step is to explore more complex prediction models such as nonlinear regression that intuitively should fit better than a simple linear regression and to contrast our prediction function with the various suggested heuristics for correcting PMI bias.

## 8 Conclusion

In this paper, we have described and compared different prediction models for the task of bilingual terminology extraction from comparable corpora. Our belief is that word co-occurrence counts prediction can be an alternative to the unreliable counts observed in small corpora. The results demonstrate the viability of the proposed approach using the PMI-Cos configuration. If more investigation is certainly needed for the Odds-Cos and Log-Jac configurations, the empirical results of our proposition suggest that predicting word co-occurrence counts is an appropriate way to improve the accuracy of the *Standard Approach* in small specialized comparable corpora.

## Acknowledgments

The research leading to these results has received funding from the French National Research Agency under grant ANR-12-CORD-0020.

## References

- Y. Cao and H. Li. 2002. Base noun phrase translation using web data and the em algorithm. *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, Taipei, Taiwan, pages 127–133.
- Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.
- Yun-Chuang Chiao and Pierre Zweigenbaum. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 1208–1212, Taipei, Taiwan.
- Béatrice Daille and Emmanuel Morin. 2005. French-English Terminology Extraction from Comparable Corpora. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCLNP'05)*, pages 707–718, Jeju Island, Korea.
- Hervé Déjean, Fatia Sadat, and Éric Gaussier. 2002. An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 218–224, Taipei, Taiwan.
- Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.
- Stefan Evert and Marco Baroni. 2007. zipfr: Word frequency modeling in r. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic.
- Robert M. Fano. 1961. *Transmission of Information: A Statistical Theory of Communications*. MIT Press, Cambridge, MA, USA.
- Pascale Fung and Kathleen Mckeown. 1997. Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora (VLC97)*, page 192202, Hong Kong.
- Pascale Fung. 1995. Compiling Bilingual Lexicon Entries From a non-Parallel English-Chinese Corpus. In David Farwell, Laurie Gerber, and Eduard Hovy, editors, *Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'95)*, pages 1–16, Langhorne, PA, USA.

- Pascale Fung. 1998. A Statistical View on Bilingual Lexicon Extraction: From ParallelCorpora to Non-parallel Corpora. In *Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'98)*, pages 1–16, Langhorne, PA, USA.
- Otero Gamallo. 2008. Evaluating two different methods for the task of extracting bilingual lexicons from comparable corpora. In *Proceedings of LREC 2008 Workshop on Comparable Corpora (LREC'08)*, pages 19–26, Marrakech, Morocco.
- I. J. Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:16–264.
- Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publisher, Boston, MA, USA.
- Clément De Groc. 2011. Babouk : Focused Web Crawling for Corpus Compilation and Automatic Terminology Extraction. In *Proceedings of The IEEE-WICACM International Conferences on Web Intelligence*, pages 497–498, Lyon, France.
- Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08)*, pages 771–779, Columbus, Ohio, USA.
- Audrey Laroche and Philippe Langlais. 2010. Revisiting Context-based Projection Methods for Term-Translation Spotting in Comparable Corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, pages 617–625, Beijing, China.
- Raghavan; Manning, Christopher D.; Prabhakar and Hinrich Schutze. 2008. Introduction to information retrieval. Cambridge University Press.
- Emmanuel Morin, Béatrice Daille, Koichi Takeuchi, and Kyo Kageura. 2007. Bilingual Terminology Mining – Using Brain, not brawn comparable corpora. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 664–671, Prague, Czech Republic.
- Viktor Pekar, Ruslan Mitkov, Dimitar Blagoev, and Andrea Mulloni. 2006. Finding translations for low-frequency words in comparable corpora. *Machine Translation*, 20(4):247–266.
- Emmanuel Prochasson and Emmanuel Morin. 2009. Anchor points for bilingual extraction from small specialized comparable corpora. *TAL*, 50(1):283–304.
- Reinhard Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pages 519–526, College Park, MD, USA.
- Raphaël Rubino and Georges Linares. 2011. A multi-view approach for term translation spotting. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'11)*, pages 29–40, Tokyo, Japan.
- Gerard Salton and Michael E. Lesk. 1968. Computer evaluation of indexing and text processing. *Journal of the Association for Computational Machinery*, 15(1):8–36.

# Measuring the Effect of Discourse Relations on Blog Summarization

**Shamima Mithun**

Concordia University  
Montreal, Quebec, Canada  
shamima.mithun@gmail.com

**Leila Kosseim**

Concordia University  
Montreal, Quebec, Canada  
kosseim@encs.concordia.ca

## Abstract

The work presented in this paper attempts to evaluate and quantify the use of discourse relations in the context of blog summarization and compare their use to more traditional and factual texts. Specifically, we measured the usefulness of 6 discourse relations - namely *comparison*, *contingency*, *illustration*, *attribution*, *topic-opinion*, and *attributive* for the task of text summarization from blogs. We have evaluated the effect of each relation using the TAC 2008 opinion summarization dataset and compared them with the results with the DUC 2007 dataset. The results show that in both textual genres, *contingency*, *comparison*, and *illustration* relations provide a significant improvement on summarization content; while *attribution*, *topic-opinion*, and *attributive* relations do not provide a consistent and significant improvement. These results indicate that, at least for summarization, discourse relations are just as useful for informal and affective texts as for more traditional news articles.

## 1 Introduction

It is widely accepted that in a coherent text, units should not be understood in isolation but in relation with each other through discourse relations that may or may not be explicitly marked. A text is not a linear combination of textual units but a hierarchical organized group of units placed together based on informational and intentional relations to one another. According to (Taboada, 2006), “Discourse relations - relations that hold together different parts (i.e. proposition, sentence, or paragraph) of the discourse - are partly responsible for the perceived coherence of a text”. For example,

in the sentence “*If you want the full Vista experience, you’ll want a heavy system and graphics hardware, and lots of memory*”, the first and second clauses do not bear much meaning independently; but become more meaningful when we realize that they are related through the discourse relation *condition*.

Discourse relations have been found useful in many NLP applications such as natural language generation (e.g. (McKeown, 1985)) and news summarization (e.g. (Blair-Goldensohn and McKeown, 2006; Bosma, 2004)) to improve coherence and better simulate human writing. However, most of these work have been developed for formal, well-written and factual documents. Text available in the social media are typically written in a more casual style, are opinionated and speculative (Andreevskaia et al., 2007). Because of this, techniques developed for formal texts, such as news articles, often do not behave as well when dealing with informal documents. In particular, news articles are more uniform in style and structure; whereas blogs often do not exhibit a stereotypical discourse structure. As a result, for blogs, it is usually more difficult to identify and rank relevant units for summarization compared to news articles.

Several work have shown that discourse relations can improve the results of summarization in the case of factual texts or news articles (e.g. (Otterbacher et al., 2002)). However, to our knowledge no work has evaluated the usefulness of discourse relations for the summarization of informal and opinionated texts, as those found in the social media. In this paper, we consider the most frequent discourse relations found in blogs: namely *comparison*, *contingency*, *illustration*, *attribution*, *topic-opinion*, and *attributive* and evaluate the effect of each relation on informal text summarization using the Text Analysis Conference (TAC)

2008 opinion summarization dataset<sup>1</sup>. We then compare these results to those found with the news articles of the Document Understanding Conference (DUC) 2007 Main task dataset<sup>2</sup>. The results show that in both types of texts, discourse relations seem to be as useful: *contingency*, *comparison*, and *illustration* relations provide a statistically significant improvement on the summary content; while the *attribution*, *topic-opinion*, and *attributive* relations do not provide a consistent and significant improvement.

## 2 Related Work on Discourse Relations for Summarization

The use of discourse relations for text summarization is not new. Most notably, (Marcu, 1997) used discourse relations for single document summarization and proposed a discourse relation identification parsing algorithm. In some work (e.g. (Bosma, 2004; Blair-Goldensohn and McKeown, 2006)), discourse relations have been exploited successfully for multi-document summarization. In particular, (Otterbacher et al., 2002) experimentally showed that discourse relations can improve the coherence of multi-document summaries. (Bosma, 2004) showed how discourse relations can be used effectively to incorporate additional contextual information for a given question in a query-based summarization. (Blair-Goldensohn and McKeown, 2006) used discourse relations for content selection and organization of automatic summaries and achieved an improvement in both cases. Discourse relations were also used successfully by (Zahri and Fukumoto, 2011) for news summarization.

However, the work described above have been developed for formal, well-written and factual documents. Most of these work show how discourse relations can be used in text summarization and show their overall usefulness. To the best of our knowledge, our work is the first to measure the effect of specific relations on the summarization of informal and opinionated text.

## 3 Tagging Discourse Relations

To evaluate the effect of discourse relations on a large scale, sentences need to be tagged automatically with discourse relations. For example, the sentence “Yesterday, I stayed at home because it

was raining.” needs to be tagged as containing a *cause* relation. One sentence can convey zero or several discourse relations. For example, the sentence “Starbucks has contributed to the popularity of good tasting coffee” does not contain any discourse relations of interest to us. On the other hand, the sentence “While I like the Zillow interface and agree it’s an easy way to find data, I’d prefer my readers used their own brain to perform a basic valuation of a property instead of relying on zestimates.” contains 5 relations of interest: one *comparison*, three *illustrations*, and one *attribution*.

### 3.1 Most Frequent Discourse Relations

Since our work is performed within the framework of blog summarization; we have only considered the discourse relations that are most useful to this application. To find the set of the relations needed for this task, we have first manually analyzed 50 summaries randomly selected from participating systems at the TAC 2008 opinion summarization track and 50 randomly selected blogs from BLOG06 corpus<sup>3</sup>. In building our relation taxonomy, we considered all main discourse relations listed in the taxonomy of Mann and Thompson’s Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). These discourse relations are also considered in Grimes’ (Grimes, 1975) and Williams’ predicate lists. From our corpus analysis, we have identified the six most prevalent discourse relations in this blog dataset, namely *comparison*, *contingency*, *illustration*, *attribution*, *topic-opinion*, and *attributive*. The *comparison*, *contingency*, and *illustration* relations are also considered by most of the work in the field of discourse analysis such as the PDTB: Penn Discourse TreeBank research group (Prasad et al., 2008) and the RST Discourse Treebank research group (Carlson and Marcu, 2001). We considered three additional classes of relations: *attributive*, *attribution*, and *topic-opinion*. These discourse relations are summarized in Figure 1 while a description of these relations is given below.

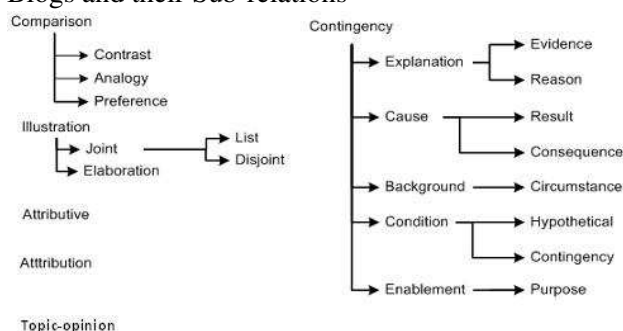
**Illustration:** Is used to provide additional information or detail about a situation. For example: “Allied Capital is a closed-end management investment company that will operate as a business development concern.”

<sup>1</sup><http://www.nist.gov/tac/>

<sup>2</sup><http://www-nlpir.nist.gov/projects/duc/guidelines/2007.html>

<sup>3</sup>[http://ir.dcs.gla.ac.uk/test\\_collections/blog06info.html](http://ir.dcs.gla.ac.uk/test_collections/blog06info.html)

Figure 1: Most Frequent Discourse Relations in Blogs and their Sub-relations



As shown in Figure 1, *illustration* relations can be sub-divided into sub-categories: *joint*, *list*, *disjoint*, and *elaboration* relations according to the RST Discourse Treebank (Carlson and Marcu, 2001) and the Penn Discourse TreeBank (Prasad et al., 2008).

**Contingency:** Provides cause, condition, reason or evidence for a situation, result or claim. For example: “*The meat is good because they slice it right in front of you.*”

As shown in Figure 1, the *contingency* relation subsumes several more specific relations: *explanation*, *evidence*, *reason*, *cause*, *result*, *consequence*, *background*, *condition*, *hypothetical*, *enablement*, and *purpose* relations according to the Penn Discourse TreeBank (Prasad et al., 2008).

**Comparison:** Gives a comparison and contrast among different situations. For example, “*Its fast-forward and rewind work much more smoothly and consistently than those of other models I’ve had.*”

The *comparison* relation subsumes the *contrast* relation according to the Penn Discourse TreeBank (Prasad et al., 2008) and the *analogy* and *preference* relations according to the RST Discourse Treebank (Carlson and Marcu, 2001).

**Attributive:** Relation provides details about an entity or an event - e.g. “*Mary has a pink coat.*”. It can be used to illustrate a particular feature about a concept or an entity - e.g. “*Picasa makes sure your pictures are always organized.*”. The *attributive* relation, also included in Grimes’ predicates (Grimes, 1975), is considered because it describes attributes or features of an object or event and is often used in query-based summarization and question answering.

**Topic-opinion:** We introduced topic-opinion relations to represent opinions which are not expressed by reported speech. This relation can be used to express an opinion: an internal feeling or belief towards an object or an event. For example: “*Cage is a wonderfully versatile actor.*”

**Attribution:** These relations are instances of reported speech both direct and indirect which may express feelings, thoughts, or hopes. For example: “*The legendary GM chairman declared that his company would make “a car for every purse and purpose.”*”

### 3.2 Automatic Discourse Tagging

Once the manual analysis identified the most prevalent set of relations, we tried to measure their frequency by tagging them automatically within a larger corpus. Only recently, the HILDA (Hernault et al., 2010) and (Feng and Hirst, 2012)’s discourse parser were made publicly available. Both of these parsers work at the text-level, as opposed to the sentence-level, and hence currently achieve the highest tagging performance when compared to the state of the art. (Feng and Hirst, 2012)’s work showed a significant improvement on the performance of HILDA by enhancing its original feature set. However, at the time this research was done, the only publicly available discourse parser was SPADE (Soricut and Marcu, 2003) which operates on individual sentences. To identify *illustration*, *contingency*, *comparison*, and *attribution* relations, we have used SPADE discourse parser. However, we have complemented this parser with three other approaches: (Jindal and Liu, 2006)’s approach is used to identify intra-sentence *comparison* relations; we have designed a tagger based on (Fei et al., 2008)’s approach to identify *topic-opinion* relations; and we have proposed a new approach to tag *attributive* relations (Mithun, 2012). A description and evaluation of these approaches can be found in (Mithun, 2012). By combining these approaches, a sentence is tagged with all possible discourse relations that it contains.

### 3.3 Distribution of Discourse Relations

To find the most prevalent discourse relations for opinion summarization, we have used the TAC 2008 opinion summarization track input document set (collection) which is a subset of BLOG06 and the answer nuggets provided by TAC 2008 as the reference summary (or model summaries), which



had been created to evaluate participants’ summaries at the TAC 2008 opinion summarization track. The collection consists of 600 blogs on 28 different topics. The dataset of the model summaries consists of 693 sentences.

Using the discourse parsers presented in Section 3.2, we computed the distribution of discourse relations within the TAC 2008 opinion summarization collection and the model summaries. *Illustration*, *contingency*, *comparison*, *attributive*, *topic-opinion*, and *attribution* are the most frequently occurring relations in our data sets. The distribution is shown in Table 1<sup>4</sup>.

Table 1: Distribution of Discourse Relations in the TAC-2008 and DUC-2007 Datasets

Discourse Relation	TAC 2008		DUC 2007	
	Coll.	Model	Coll.	Model
Illustration	52%	46%	42%	38%
Contingency	31%	37%	34%	29%
Comparison	23%	18%	15%	12%
Attributive	12%	28%	3%	4%
Topic-opinion	14%	15%	4%	5%
Attribution	11%	9%	2%	3%
other	13%	9%	28%	31%
none	14%	10%	8%	7%

Table 1 shows that in the TAC 2008 input document set, the *illustration* relation occurs in 52% of the sentences; while *attribution* is the least frequently occurring relation. In this dataset, other relations, such as *antithesis* and *temporal* relations, occur in about 13% of the sentences and about 14% of the sentences did not receive any relation tag. As indicated in Table 1, the TAC model summaries have a similar distribution as the collection as a whole. The *attributive* relation seems, however, to be more frequent in the summaries (28%) than in the original texts (12%). We suspect that the reason for this is due to the question types of this track. To successfully generate query-relevant summaries that answer the questions of this track, candidate sentences need to contain *attributive* relations. For example, to answer the questions from this track “*Why do people like Picasa?*” or “*What features do people like about Windows Vista?*”, the summary needs to provide details about these entities or illustrate a particular feature about them. As a result, the summary will be composed of many *attributive* relations since

<sup>4</sup>In Table 1, the percentages do not add up to 100 because a sentence may contain more than one relation.

*attributive* relations help to model the required information.

To compare the distribution of discourse relations within more formal types of texts such as news articles, we used the Document Understanding Conference (DUC) 2007 Main Task input document set (collection) and their associated model summaries. The DUC 2007 dataset is a news article based dataset from the AQUAINT corpus. The DUC 2007 input document set contains 1125 news articles on 45 different topics. The model summaries were used to evaluate the DUC 2007 participants’ summaries. The dataset of the model summaries contains 180 summaries generated by the National Institute of Standards and Technology (NIST) assessors with a summary length of about 250 words. The distribution of relations in this dataset are shown in Table 1.

Table 1 shows that the most frequently occurring relation in the DUC 2007 document collection and in the model summaries is *illustration*; while the *attribution* relation is the least frequently occurring relation. Here again, it is interesting to note that the distribution of the discourse relations in the document collection and in the model summaries is generally comparable.

The distribution of the *illustration*, *contingency*, and *comparison* relations in the DUC 2007 dataset is comparable to those in the TAC 2008 opinion summarization dataset. Indeed, Table 1 shows that *illustration*, *contingency*, and *comparison* relations occur quite frequently irrespective of the textual genre. However, in contrast to the TAC dataset, *attributive*, *topic-opinion*, and *attribution* relations occur very rarely in DUC 2007. We suspect that this is mostly due to the opinionated nature of blogs. Another observation is that *temporal* relations (included in “other”) occurred very frequently (30%) in the DUC 2007 dataset whereas this relation occurs rarely in the blog dataset. This is inline with our intuition that news articles present events that inherently contain temporal information.

## 4 Evaluation of Discourse Relations

To measure the usefulness of discourse relations for the summarization of informal texts, we have tested the effect of each relation with four different summarizers: BlogSum (Mithun, 2012), MEAD (Radev et al., 2004), the best scoring sys-

tem at TAC 2008<sup>5</sup> and the best scoring system at DUC 2007<sup>6</sup>. We have evaluated the effect of each discourse relation on the summaries generated and compared the results. Let us first describe the BlogSum summarizer.

#### 4.1 BlogSum

BlogSum is a domain-independent query-based extractive summarization system that uses intra-sentential discourse relations within the framework based on text schemata. The heart of BlogSum is based on discourse relations and text schemata.

BlogSum works in the following way: First candidate sentences are extracted and ranked using the topic and question similarity to give priority to topic and question relevant sentences. Since BlogSum has been designed for blogs, which are opinionated in nature, to rank a sentence, the sentence polarity (e.g. positive, negative or neutral) is calculated and used for sentence ranking. To extract and rank sentences, BlogSum thus calculates a score for each sentence using the features shown below:

$$\text{Sentence Score} = w_1 \times \text{Question Similarity} + w_2 \times \text{Topic Similarity} + w_3 \times \text{Subjectivity Score}$$

where, question similarity and topic similarity are calculated using the cosine similarity based on words *tf.idf* and the subjectivity score is calculated using a dictionary-based approach based on the MPQA lexicon<sup>7</sup>. Once sentences are ranked, they are categorized based on the discourse relations that they convey. This step is critical because the automatic identification of discourse relations renders BlogSum independent of the domain. This step also plays a key role in content selection and summary coherence as schemata are designed using these relations.

In order not to answer all questions the same way, BlogSum uses different schemata to generate a summary that answers specific types of questions. Each schema is designed to give priority to its associated question type and subjective sentences as summaries for opinionated texts are generated. Each schema specifies the types of discourse relations and the order in which they should appear in the output summary for a par-

ticular question type. Figure 2 shows a sample schema that is used to answer *reason* questions (e.g. “*Why do people like Picasa?*”). According to this schema<sup>8</sup>, one or more sentences containing a *topic-opinion* or *attribution* relation followed by zero or many sentences containing a *contingency* or *comparison* relation followed by zero or many sentences containing a *attributive* relation should be used.

Figure 2: A Sample Discourse Schema used in BlogSum

Relations & Constraints	
Relation:	{ <i>Topic-opinion/ Attribution</i> } <sup>+</sup>
Constraint:	Sentence Polarity.
Relation:	{ <i>Contingency/ Comparison</i> } <sup>*</sup>
Constraint:	Compared Objects, Sentence Focus.
Relation:	<i>Attributive</i> <sup>*</sup>
Constraint:	Sentence Focus.

Finally the most appropriate schema is selected based on a given question type; and candidate sentences fill particular slots in the selected schema based on which discourse relations they contain in order to create the final summary (details of BlogSum can be found in (Mithun, 2012)).

#### 4.2 Evaluation of Discourse Relations on Blogs

To evaluate the effect of each discourse relation for blog summarization, we performed several experiments. We used as a baseline the original ranked list of candidate sentences produced by BlogSum before applying the discourse schemata, and compared this to the BlogSum-generated summaries with and without each discourse relation. We used the TAC 2008 opinion summarization dataset which consists of 50 questions on 28 topics; on each topic one or two questions were asked and 9 to 39 relevant documents were given. For each question, one summary was generated with no regards to discourse relations and two summaries were produced by BlogSum: one using the discourse tagger and the other without using the specific discourse tagger. The maximum summary length was restricted to 250 words.

To measure the effect of each relation, we have automatically evaluated how BlogSum performs using the standard ROUGE-2 and ROUGE-SU4

<sup>5</sup><http://www.nist.gov/tac/>

<sup>6</sup><http://www-nlpir.nist.gov/projects/duc/guidelines/2007.html>

<sup>7</sup>MPQA: <http://www.cs.pitt.edu/mpqa>

<sup>8</sup>The notation / indicates an alternative, { } indicates optionality, \* indicates that the item may appear 0 to n times and + indicates that the item may appear 1 to n times

measures. For comparative purposes, Table 2 shows the official ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) for all 36 submissions of the TAC 2008 opinion summarization track. In the table, “TAC Average” refers to the mean performance of all participant systems and “TAC-Best” refers to the best-scoring system at TAC 2008.

Table 2: Results of the TAC 2008 Opinion Summarization Track

System Name	R-2	R-SU4
TAC Average	0.069	0.086
TAC-Best	0.130	0.139

Table 3: Effect of Discourse Relations on ROUGE-2 with the TAC 2008 Dataset

System Name	BlogSum R-2	MEAD R-2	TAC-Best R-2
Baseline	0.102↓	0.041↓	0.130
w/o Illustration	0.107↓	0.022↓	0.112↓
w/o Contingency	0.093↓	0.025↓	0.102↓
w/o Comparison	0.103↓	0.033↓	0.113↓
w/o Attributive	0.113↓	0.050	0.124
w/o Topic-opinion	0.112↓	0.049	0.123
w/o Attribution	0.118↓	0.051↓	0.128
with all Relations	<b>0.125</b>	<b>.053</b>	<b>0.138</b>

Table 4: Effect of Discourse Relations on ROUGE-SU4 with the TAC 2008 Dataset

System Name	BlogSum R-SU4	MEAD R-SU4	TAC-Best R-SU4
Baseline	0.107↓	0.064↓	0.139
w/o Illustration	0.110↓	0.041↓	0.120↓
w/o Contingency	0.102↓	0.046↓	0.110↓
w/o Comparison	0.108↓	0.052↓	0.122↓
w/o Attributive	0.115↓	0.072	0.130
w/o Topic-opinion	0.117	0.072	0.129
w/o Attribution	0.127↓	0.073↓	0.132
with all Relations	<b>0.128</b>	<b>0.075</b>	<b>0.151</b>

The results of our evaluation are shown in Tables 3 (ROUGE-2) and 4 (ROUGE-SU4). As the tables show, BlogSum’s baseline is situated below the best scoring system at TAC-2008, but much higher than the average system (see Table 2); hence, it represents a fair baseline. The tables further show that using both the ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) metrics, with the TAC 2008 dataset, BlogSum performs better when taking discourse relations into account. Indeed, when ignoring discourse relations, BlogSum has a R2=0.102 and R-SU4=0.107 and misses many question relevant sentences; whereas the inclusion of these relations helps to incorporate those relevant sentences into the final summary and brings

the R-2 score to 0.125 and R-SU4 to 0.128. In order to verify if these improvements were statistically significant, we performed a 2-tailed t-test. The results of this test are indicated with the ↓ symbol in Tables 3 and 4. For example, the baseline setup of BlogSum performed significantly lower for both R-2 and R-SU4 compared to BlogSum with all relations. This result indicates that the use of discourse relations as a whole helps to include more question relevant sentences and improve the summary content.

To ensure that the results were not specific to our summarizer, we performed the same experiments with two other systems: the MEAD summarizer (Radev et al., 2004), a publicly available and a widely used summarizer, and with the output of the TAC best-scoring system. For MEAD, we first generated candidate sentences using MEAD, then these candidate sentences were tagged using discourse relation taggers used under BlogSum. Then these tagged sentences were filtered using BlogSum so that no sentence with a specific relation is used in summary generation for a particular experiment. We have calculated ROUGE scores using the original candidate sentences generated by MEAD and also using the filtered candidate sentences. As a baseline, we used the original candidate sentences generated by MEAD. As a best case scenario, we have passed these candidate sentences through the discourse schemata used by BlogSum (see Section 4.1). In Tables 3 and 4, this is referred to as “MEAD with all relations”. We have applied the same approach with the output of the TAC best-scoring system. In the tables, “TAC-Best Baseline” refers to the original summaries generated by the TAC-Best system and “TAC-Best with all relations” refers to the summaries generated by applying discourse schemata using the summary sentences generated by the TAC-Best system.

When looking at individual relations, Tables 3 and 4 show that considering *illustrations*, *contingencies* and *comparisons* make a statistically significant improvement in all scenarios, and with all summarisers. For example, if TAC-Best does not consider *illustration* relations, then the R-2 score decreases from 0.138 to 0.112, 0.102 and 0.113, respectively. On the other hand, the relations of *topic-opinion*, *attribution*, and *attributive* do not consistently lead to a statistically significant improvement on ROUGE scores.

It is interesting to note that although informal texts may not exhibit a clear discourse structure, the use of individual discourse relations such as *illustration*, *contingency* and *comparison* is nonetheless useful in the analysis of informal documents such as those found in the social media.

### 4.3 Effect of Discourse Relations on News

To compare the results found with blogs with more formal types of texts, we have performed the same experiments but, this time with the DUC 2007 Main Task dataset. In this task, given a topic (title) and a set of 25 relevant documents, participants had to create an automatic summary of length 250 words from the input documents. In the dataset, there were 45 topics and thirty teams participated to this shared task. Table 5 shows the official ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores of the DUC 2007 main task summarization track. In Table 5, “DUC Average” refers to the mean performance of all participant systems and “DUC-Best” refers to the best scoring system at DUC 2007.

Table 5: DUC 2007 Main Task Summarization Results

System Name	R-2	R-SU4
DUC Average	0.095	0.157
DUC-Best	0.124	0.177

Table 6: Effect of Discourse Relations on ROUGE-2 with the DUC 2007 Dataset

System Name	BlogSum R-2	MEAD R-2	DUC-Best R-2
Baseline	0.089	0.099	0.124↓
w/o Illustration	0.079↓	0.061↓	0.103↓
w/o Contingency	0.074↓	0.060↓	0.097↓
w/o Comparison	0.086↓	0.078↓	0.114↓
w/o Attributive	0.092	0.099	0.119↓
w/o Topic-opinion	0.092	0.099	0.115↓
w/o Attribution	0.093	0.099	0.120↓
with all Relations	<b>0.093</b>	<b>0.110</b>	<b>0.157</b>

Tables 6 and 7 show the results with this dataset with respect to ROUGE-2 and ROUGE-SU4, respectively. As the tables show, BlogSum’s performance with all discourse relations (R2=0.093 and R-SU4=0.132) is similar to the DUC average performance shown in Table 5 (R2=0.095 and R-SU4=0.157) which is much

Table 7: Effect of Discourse Relations on ROUGE SU-4 with the DUC 2007 Dataset

System Name	BlogSum R-SU4	MEAD R-SU4	DUC-Best R-SU4
Baseline	0.110↓	0.142↓	0.177↓
w/o Illustration	0.117↓	0.118↓	0.138↓
w/o Contingency	0.113↓	0.118↓	0.123↓
w/o Comparison	0.122↓	0.130↓	0.144↓
w/o Attributive	0.131	0.141↓	0.159↓
w/o Topic-opinion	0.130	0.141↓	0.153↓
w/o Attribution	0.131	0.142↓	0.164↓
with all Relations	<b>0.132</b>	<b>0.168</b>	<b>0.196</b>

lower than the DUC-Best performance (R2=0.124, R-SU4=0.177) shown in Table 5). However, these results show that even though BlogSum was designed for informal texts, it still performs relatively well with formal documents. Tables 6 and 7 further show that with the news dataset, the same relations have the most effect as with blogs. Indeed BlogSum generated summaries also benefit most from the *contingency*, *illustration*, and *comparison* relations; and all three relations bring a statistically significant contribution to the summary content.

Here again, as shown in Tables 6 and 7, we performed the same experiments with two other systems: the MEAD summarizer and the output of the DUC-Best system. Again, for the DUC 2007 dataset, each discourse relation has the same effect on summarization with all systems as with the blog dataset: *contingency*, *illustration*, and *comparison* provide a statistically significant improvement in content; while *attributive*, *topic-opinion* and *attribution* do not reduce the content, but do not see to bring a systematic and significant improvement.

## 5 Conclusion and Future Work

In this paper, we have evaluated the effect of discourse relations on summarization. We have considered the six most frequent relations in blogs - namely *comparison*, *contingency*, *illustration*, *attribution*, *topic-opinion*, and *attributive*. First, we have measured the distribution of discourse relations on blogs and on news articles and show that the prevalence of these six relations is not genre dependent. For example, the relations of *illustration*, *contingency*, and *comparison* occur frequently in both textual genres. We have then evaluated the effect of these six relations on summa-

rization with the TAC 2008 opinion summarization dataset and the DUC 2007 dataset. We have conducted these evaluations with our summarization system called BlogSum, the TAC best-scoring system, the DUC best-scoring system, and the MEAD summarizer. The results show that for both textual genres, some relations have more effect on summarization compared to others. In both types of texts, the *contingency*, *illustration*, and *comparison* relations provide a significant improvement on summary content; while the *attribution*, *topic-opinion*, and *attributive* relations do not provide a systematic and statistically significant improvement. These results seem to indicate that, at least for summarization, discourse relations are just as useful for informal and affective texts as for more traditional news articles. This is interesting, because although informal texts may not exhibit a clear discourse structure, the use of individual discourse relations is nonetheless useful in the analysis of informal documents.

In the future, it would be interesting to evaluate the effect of other relations such as the *temporal* relation. Indeed, *temporal* relations occur infrequently in blogs but are very frequent in news articles. Such an analysis would allow us to tailor the type of discourse relations to include in the final summary as a function of the textual genre being considered. In the future, it would also be interesting to use other types of texts such as reviews and evaluate the effect of discourse relations using other measures than ROUGE-2 and ROUGE-SU4. Finally, we would like to validate this work again with the newly available discourse parsers of (Hernault et al., 2010) and (Feng and Hirst, 2012).

## Acknowledgement

The authors would like to thank the anonymous referees for their valuable comments on an earlier version of the paper. This work was financially supported by an NSERC grant.

## References

Andreevskaia, A., Bergler, S., Urseanu, M.: All Blogs are Not Made Equal: Exploring Genre Differences in Sentiment Tagging of Blogs. *In Proceedings of the International Conference on Weblogs and Social Media (ICWSM-2007)*, (2007), Boulder, Colorado.

Blair-Goldensohn, S.J., McKeown, K.: Integrating Rhetorical-Semantic Relation Models for Query-Focused Summarization. *In Proceedings of the Doc-*

*ument Understanding Conference (DUC) Workshop at NAACL-HLT 2006*, (2006), New York, USA.

- Bosma, W.: Query-Based Summarization using Rhetorical Structure Theory. *In Proceedings of the 15th Meeting of Computational Linguistics in the Netherlands CLIN*, (2004), Leiden, Netherlands.
- Carlson, L., Marcu, D.: Discourse Tagging Reference Manual. University of Southern California Information Sciences Institute, ISI-TR-545, 2001.
- Fei, Z., Huang, X., Wu, L.: Mining the Relation between Sentiment Expression and Target Using Dependency of Words. *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, 257–264 (2008), Wuhan, China.
- Feng, V. W., Hirst, G.: Text-level Discourse Parsing with Rich Linguistic Features. *In Proceedings of ACL-2012*, 60–68 (2012), Stroudsburg, USA.
- Grimes, J. E.: The Thread of Discourse. Technical report No. NSF-TR-1, NSF-GS-3180. Cornell University, Ithaca, New York, 1975.
- Hernault, H., Prendinger, H., duVerle, D. A., Ishizuka, M.: HILDA: A discourse parser using support vector machine classification. *J. Dialogue and Discourse*, 1(3):1–33, 2010.
- Jindal, N., Liu, B.: Identifying Comparative Sentences in Text Documents. *In Proceedings of SIGIR-2006*, 244–251 (2006), Washington, USA.
- Mann, W.C., Thompson, S. A.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organisation. *J. Text*, 3(8):234–281, 1988.
- Marcu, D.: From Discourse Structures to Text Summaries. *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*. 1997, 82–88, Madrid, Spain.
- McKeown, K.R.: Discourse Strategies for Generating Natural-Language Text. *J. Artificial Intelligence*, 27(1):1–41, 1985.
- Mithun, S.: Exploiting Rhetorical Relations in Blog Summarization. *PhD Thesis*, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, 2012.
- Otterbacher, J. C., Radev, D. R., Luo, A.: Revisions that Improve Cohesion in Multi-document Summaries: A Preliminary Study. *In Proceedings of the ACL-2002 Workshop on Automatic Summarization*, 27–36 (2002), Philadelphia, USA.
- Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., Webber, B.: The Penn Discourse Treebank 2.0. Annotation Manual. University of Pennsylvania, IRCS-08-01, 2008.
- Radev, D. et al.: MEAD -A Platform for Multidocument Multilingual Text Summarization. *In Proceedings of LREC-2004*, 1–4 (2004), Lisbon, Portugal.

- Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. *In Proceedings of NAACL/HLT 2003*, 149–156 (2003), Edmonton, Canada.
- Taboada, M.: Discourse Markers as Signals (or not) of Rhetorical Relations. *J. Pragmatics*, 38(4):567–592, 2006.
- Zahri, N. A. H. B., Fukumoto, F.: Multi-document Summarization Using Link Analysis Based on Rhetorical Relations between Sentences. *In Proceedings of CICLing*, 328–338 (2011), Tokyo, Japan.

# Supervised Sentence Fusion with Single-Stage Inference

Kapil Thadani and Kathleen McKeown

Department of Computer Science

Columbia University

New York, NY 10025, USA

{kapil, kathy}@cs.columbia.edu

## Abstract

Sentence fusion—the merging of sentences containing similar information—has been shown to be useful in an abstractive summarization context. We present a new dataset of sentence fusion instances obtained from evaluation datasets in summarization shared tasks and use this dataset to explore supervised approaches to sentence fusion. Our proposed inference approach recovers the highest scoring output fusion under an n-gram factorization using a compact integer linear programming formulation that avoids cycles and disconnected structures. In addition, we introduce simple fusion-specific features and constraints that outperform a compression-inspired baseline as well as a variant that relies on human-identified concept spans for perfect content selection.

## 1 Introduction

Abstractive text summarization has long been a high-level goal of natural language processing. Although progress in text-to-text (T2T) generation tasks such as sentence compression and paraphrase generation has been steady, the fusion of multiple sentences offers a particularly formidable challenge. Sentence fusion refers to the task of combining two or more sentences which overlap in information content, avoiding extraneous details and preserving common information. This procedure has been observed in human summarization (Jing and McKeown, 2000) and has been shown to be a valuable component of automated summarization systems (Barzilay and McKeown,

2005). However, research in sentence fusion has long been hampered by the absence of datasets for the task, and the difficulty of generating one has cast doubt on the viability of automated fusion (Daumé III and Marcu, 2004).

This paper presents a new fusion dataset generated from existing human annotations and also introduces a discriminative T2T system that generalizes the single sentence compression approach of Thadani and McKeown (2013) to n-way sentence fusion. Our fusion dataset is constructed from evaluation data for summarization shared tasks in the Document Understanding Conference (DUC)<sup>1</sup> and the Text Analysis Conference (TAC).<sup>2</sup> Specifically, we use human-generated annotations produced for the pyramid method (Nenkova et al., 2007) for summarization evaluation to produce a dataset of natural human fusions with quantifiable agreement. This offers advantages over previous datasets used for standalone English sentence fusion which contain annotator-induced noise (McKeown et al., 2010) or cannot be distributed (Elsner and Santhanam, 2011). In addition, both these datasets contain approximately 300 instances of fusion while the new dataset presented here contains 1858 instances.

Crucially, this larger corpus encourages supervised approaches to sentence fusion and we leverage this to explore new strategies for the task. Previous approaches to fusion have generally relied on variations of dependency graph combination (Barzilay and McKeown, 2005; Filippova and Strube, 2008b; Elsner and Santhanam, 2011) for content selection with a separate step for linearization that is usually based on a language model (LM). In contrast, we experiment with combin-

<sup>1</sup><http://duc.nist.gov>

<sup>2</sup><http://www.nist.gov/tac>

1	In 1991, the independents claimed nearly a third of adult book purchases <b>but six years later their market share</b> was nearly cut in half, <b>down to 17%</b> .
2	<b>By 1999, independent booksellers held only a 17 percent market share.</b>
SCU	Six years later independent booksellers' market share was down to 17%
1	<b>The heavy-metal group Metallica filed a federal lawsuit in 2000 against Napster for copyright infringement</b> , charging that Napster encouraged users to trade copyrighted material without the band's permission.
2	The heavy metal rock band <b>Metallica</b> , rap artist Dr. Dre and the RIAA <b>have sued Napster</b> , developer of Internet sharing software, <b>alleging the software enables the acquisition of copyrighted music without permission.</b>
3	<b>The heavy-metal band Metallica sued Napster</b> and three universities <b>for copyright infringement</b> and racketeering, seeking \$10 million in damages.
SCU	Metallica sued Napster for copyright infringement
1	The government was to pardon 23 FARC members as the two sides <b>negotiate prisoner exchanges.</b>
2	The Columbian government plans to pardon more than 30 members of FARC as <b>they negotiate a prisoner swap.</b>
3	<b>The government and FARC continued to argue over details of a prisoner swap.</b>
SCU	The government and FARC negotiate prisoner exchanges

Table 1: SCU annotations drawn from DUC 2005–2007 and TAC 2008–2011. Human-annotated contributors to the SCU are indicated as boldfaced spans within the respective source sentences.

ing linearization with content selection to produce a single-stage joint approach to fusion. For this, we adapt the sequential *structured transduction* approach described in Thadani and McKeown (2013) for sentence compression and extend it to process multiple input sentences for fusion tasks. This discriminative approach to sentence generation permits rich features that estimate the informativeness of specific tokens chosen from the input sentences as well as the fluency of the n-grams used to assemble them for the output sentence. Furthermore, our inference formulation allows all potential orderings of input tokens to be considered in the output and prevents degenerate cyclic or disjoint orderings via *commodity flow* constraints (Magnanti and Wolsey, 1994).

The primary contributions of this work are:

- A novel dataset of natural sentence fusions drawn from a corpus of pyramid evaluations for summarization shared tasks which is available to the NLP community.
- A supervised approach to sentence fusion that jointly addresses non-redundant content selection and linearization.

We evaluated the proposed fusion system against a basic compression baseline that does not include fusion-specific features as well as a proposed strong baseline that directly leverages human-annotated concept boundaries in the original dataset, thereby avoiding the issue of content selection. An evaluation under a variety of automated metrics indicates that our proposed approach strongly outperforms the former and appears competitive with the latter.

## 2 Pyramid fusion corpus

The pyramid method is a technique for summarization evaluation that aims to quantify the semantic content of summaries and compare automated summaries to human summaries on the basis of this semantic content (Nenkova et al., 2007). For each summarization topic to be evaluated, a number of human-authored summaries are first produced. In the DUC and TAC evaluations, the number of summaries is usually fixed at 7 per topic. A collection of *summarization content units* or SCUs—intended to correspond to atomic units of information—are then generated by annotators reading these summaries. Each SCU comprises a *label* which is a concise English sentence that states the meaning of the SCU<sup>3</sup> and a list of *contributors* which are discontinuous character spans from the summary sentences—hereafter referred to as *source* sentences—in which that SCU is realized. Table 1 contains examples of SCUs drawn from DUC 2005–2007 and TAC 2008–2011 data.

Our fusion corpus is constructed by taking the source sentences of an SCU as input and the SCU labels as the gold fusion output. The fusion task posed by this corpus is similar to sentence *intersection* as defined by Marsi and Kraemer (2005) although it does not fit the criteria for *strict* intersection as addressed in Thadani and McKeown (2011) since source sentences do not always expressly mention all the information in an SCU label due to unresolved anaphora and entail-

<sup>3</sup>An SCU annotation guide from DUC 2005 is available at <http://www1.cs.columbia.edu/~ani/DUC2005/AnnotationGuide.htm>.



ment. The following procedure was used to extract meaningful fusion instances from the SCUs.

1. SCUs that have no more than one contributor which covers a single summary sentence are dropped. In addition, we chose to restrict the number of input sentences to at most four<sup>4</sup> since larger SCUs are very infrequent.
2. Although SCU descriptions are required to be full sentences, we found that this was not upheld in practice. We therefore removed SCUs whose labels contain fewer than 5 words and did not have an identifiable verb beyond the first token. As a practical consideration, SCUs with source sentences which have more than 100 tokens were also dropped.
3. Annotated concepts in this dataset often only cover a small fraction of source sentences and may not represent the full overlap between them. To account for this, we ignored SCUs without contributors that are at least half the length of their source sentences as well as SCUs whose labels are less than half the length of the smallest contributor.
4. Finally, we chose to retain only SCUs whose labels contain terms present in at least one source sentence, thus ensuring that gold fusions are reachable without paraphrasing.

This yields 1858 fusion instances of which 873 have two inputs, 569 have three and 416 have four.

### 3 Single-stage Fusion

Previous approaches to fusion have often relied on dependency graph combination (Barzilay and McKeown, 2005; Filippova and Strube, 2008b; Elsner and Santhanam, 2011) to produce an intermediate syntactic representation of the information in the sentence. Linearization of output fusions is usually performed by ranking hypotheses with a language model (LM), sometimes with language-specific heuristics to filter out ill-formed sentences. This approach is also known as *overgenerate-and-rank* and is often found to be a source of errors in T2T problems (Barzilay and McKeown, 2005).

Although syntactic representations are natural for assembling text across sentences, recent work in unsupervised multi-sentence fusion has shown that well-formed output can often be constructed

<sup>4</sup>This is accomplished by removing additional contributors that share the fewest words with the SCU label.

purely on the basis of adjacency relationships in a word graph (Filippova, 2010). Similarly, systems for related T2T tasks such as sentence compression (McDonald, 2006; Clarke and Lapata, 2008) and strict sentence intersection (Thadani and McKeown, 2011) have also seen promising results by linearizing n-grams without explicitly relying on syntactic representations.

Our framework takes a similar perspective and assembles output text directly from n-grams over input tokens, but we employ a discriminative structured prediction approach in which likelihood under an LM is one of many features of output quality and parameters for all features are learned from a training corpus. Moreover, rather than rely on pipelined stages to first select the output content and then linearize an intermediate representation, we jointly address token selection alongside phrase-based ordering thereby yielding a single-stage approach to fusion.

#### 3.1 ILP formulation

The starting point for this work is the sequential structured transduction<sup>5</sup> model of Thadani and McKeown (2013), originally devised for single sentence compression. This approach relies on integer linear programming (ILP) to find a globally optimal solution to generation problems involving heterogeneous substructures. ILP has been used frequently in recent T2T generation systems including many for sentence fusion (Filippova and Strube, 2008b; Elsner and Santhanam, 2011), intersection (Thadani and McKeown, 2011) and compression (Clarke and Lapata, 2008; Filippova and Strube, 2008a; Berg-Kirkpatrick et al., 2011), as well as other natural language processing tasks. Although LPs with integer constraints are NP-hard in the general case, the availability of optimized general-purpose ILP solvers and the natural limits on English sentence length make ILP inference attractive for sentence-level optimization problems.

Consider a single fusion instance involving  $k$  source sentences  $\mathcal{S} \triangleq \{S_1, \dots, S_k\}$ . The notation  $F_{\mathcal{S}}$  is used to denote a fusion of the sentences in  $\mathcal{S}$ . The inference step aims to retrieve the output sentence  $F_{\mathcal{S}}^*$  that is the most likely fusion of  $\mathcal{S}$ , i.e., the sentence that maximizes  $p(F_{\mathcal{S}}|\mathcal{S})$  or equivalently maximizes some scoring function  $score(F_{\mathcal{S}})$ . In

<sup>5</sup>The full joint model presented in Thadani and McKeown (2013) also explicitly infers tree-structured dependencies, but we found in preliminary experiments that this did not perform well with multiple sentence inputs. See discussion in §5.

our feature-based discriminative setting, we define  $score(F_S)$  as a dot product of weights  $\mathbf{w}$  and a feature map  $\Phi(S, F_S)$  defined over the fusion and its input; in other words

$$F_S^* \triangleq \arg \max_{F_S} \mathbf{w}^\top \Phi(S, F_S) \quad (1)$$

The feature map  $\Phi$  for an arbitrary fusion sentence is defined to factor over the words and potential n-grams from the input text. Let  $T \triangleq \{t_i : 1 \leq i \leq N_j, 1 \leq j \leq |\mathcal{S}|\}$  represent the set of tokens (including duplicates) in  $\mathcal{S}$  and let  $x_i \in \{0, 1\}$  represent a token indicator variable whose value corresponds to whether token  $t_i$  is present in the output sentence  $F_S$ . We also consider n-gram phrases defined over the tokens in  $T$  and assume the use of bigrams without loss of generality.<sup>6</sup> Let  $U$  represent the set of all possible bigrams that can be constructed from the tokens in  $T$ ; in other words  $U \triangleq \{(t_i, t_j) : t_i \in T \cup \{\text{START}\}, t_j \in T \cup \{\text{END}\}, i \neq j\}$ . Following the notation for token indicators, let  $y_{ij} \in \{0, 1\}$  represent a bigram indicator variable for whether the contiguous pair of tokens  $\langle t_i, t_j \rangle$  is in the output sentence. We represent entire token and bigram configurations with incidence vectors  $\mathbf{x} \triangleq \langle x_i \rangle_{t_i \in T}$  and  $\mathbf{y} \triangleq \langle y_{ij} \rangle_{\langle t_i, t_j \rangle \in U}$  which are equivalent to some subset of  $T$  and  $U$  respectively. With this notation, (1) can be rewritten as

$$\begin{aligned} F_S^* &= \arg \max_{\mathbf{x}, \mathbf{y}} \sum_{t_i \in T} x_i \cdot \mathbf{w}_{\text{tok}}^\top \phi_{\text{tok}}(t_i) \\ &\quad + \sum_{\langle t_i, t_j \rangle \in U} y_{ij} \cdot \mathbf{w}_{\text{ngr}}^\top \phi_{\text{ngr}}(\langle t_i, t_j \rangle) \\ &= \arg \max_{\mathbf{x}, \mathbf{y}} \mathbf{x}^\top \boldsymbol{\theta}_{\text{tok}} + \mathbf{y}^\top \boldsymbol{\theta}_{\text{ngr}} \end{aligned} \quad (2)$$

where  $\phi$  is a feature vector for tokens or bigrams and  $\mathbf{w}$  is a corresponding vector of weight parameters. Each  $\boldsymbol{\theta} \triangleq \langle \mathbf{w}^\top \phi(s) \rangle$  is therefore a vector of feature-based scores for either tokens or bigrams.

The joint objective in (2) conveniently permits content-based features in  $\phi_{\text{tok}}$  for content selection and fluency features such as LM log-likelihoods in  $\phi_{\text{ngr}}$  for linearization. However, decoding a valid sentence with this objective is non-trivial. Merely selecting the tokens and bigrams that maximize (2) is liable to produce degenerate structures, i.e., cycles, disconnected components, branches and inconsistency between the token and bigram configurations in  $\mathbf{x}$  and  $\mathbf{y}$ . Most prior T2T linearization

<sup>6</sup>This approach permits n-grams of any order (Thadani and McKeown, 2013) but we use bigrams here to produce ILPs that scale quadratically with the number of input tokens.

approaches such as the Viterbi-based approach of McDonald (2006) and the ILP of Clarke and Lapata (2008) cannot be applied when the tokens in the input do not have a total ordering, as is the case when the input consists of more than one sentence.

### 3.2 Structural Constraints

We now briefly describe the structural constraints proposed by Thadani and McKeown (2013) to address the problem of degeneracy in sentential structure. First, we consider the problem of output *consistency*—more formally, bigram variables  $y_{ij}$  that are non-zero must activate their token variables  $x_i$  and  $x_j$  while token variables can only activate a single bigram variable in the first and second position each.

$$x_i - \sum_j y_{ij} = 0, \quad \forall t_j \in T \quad (3)$$

$$x_j - \sum_i y_{ij} = 0, \quad \forall t_i \in T \quad (4)$$

The second requirement for non-degenerate output is that non-zero  $y_{ij}$  must form a sentence-like *linear ordering* of tokens, avoiding cycles and branching. For this purpose, auxiliary variables are introduced to establish *single-commodity flow* (Magnanti and Wolsey, 1994) between all pairs of tokens that may appear adjacent in the output. Linear token ordering is maintained by defining real-valued commodity flow variables  $\gamma_{ij}$  which are non-negative.

$$\gamma_{ij} \geq 0, \quad \forall \langle t_i, t_j \rangle \in U \quad (5)$$

Each active token in the solution must have some positive incoming commodity and *consumes* one unit of this commodity, transmitting the remaining value to outgoing flow variables. This ensures that cycles cannot be present in the flow structure.

$$\sum_i \gamma_{ij} - \sum_k \gamma_{jk} = x_j, \quad \forall t_j \in T \quad (6)$$

The acyclic flow structure can be imparted to  $\mathbf{y}$  by constraining bigram indicators to be active only if their corresponding tokens have positive commodity flow between them.

$$\gamma_{ij} - C_{\max} y_{ij} \leq 0, \quad \forall \langle t_i, t_j \rangle \in U \quad (7)$$

where  $C_{\max}$  is the maximum amount of commodity that the  $\gamma_{ij}$  variables may carry and serves as an upper bound on the number of output tokens.

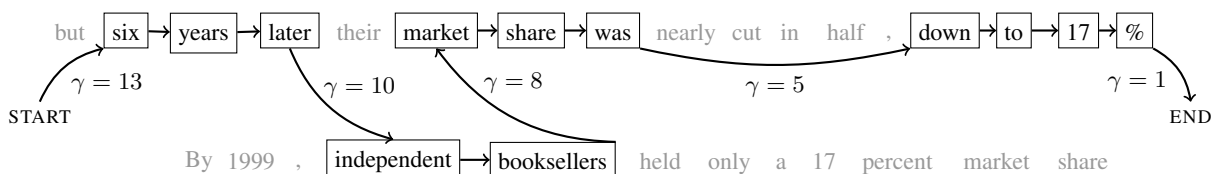


Figure 1: An illustration of commodity values for a valid solution of the ILP.

Finally, in order to establish *connectivity* in the output, we also introduce indicator variables  $y_{*j}$  and  $y_{i*}$  to denote the sentence-starting and terminating bigrams  $\langle \text{START}, t_j \rangle$  and  $\langle t_i, \text{END} \rangle$  respectively. A valid output sentence must be started and terminated by exactly one bigram.

$$\sum_j y_{*j} = 1 \quad (8)$$

$$\sum_i y_{i*} = 1 \quad (9)$$

Flow variables  $\gamma_{*j}$  and  $\gamma_{i*}$ , are also defined for START and END respectively. Since START has no incoming flow variables, the amount of commodity in  $\gamma_{*j}$  are unconstrained. This provides the only point of origin for the commodity and, in conjunction with (7), induces connectivity in  $\mathbf{y}$ .

### 3.3 Further Extensions for Fusion

The constraints specified above are adequate to enforce structural soundness in an output sentence and are applicable to a range of T2T linearization problems. We now address the issue of *redundancy*, which is unique to sentence fusion. The input sentences are expected to contain overlapping information which is useful to identify because: (a) it is a signal of salience, and (b) it is reasonable to expect that this repeated information should not appear redundantly in the output.

#### 3.3.1 Supported content words

To address the first point above, we iterate through each sentence and generate groups  $G$  of similar or identical tokens across sentences, which we refer to as *supported* tokens. The selection of tokens is limited to open-class words such as nouns, verbs, adjectives and adverbs. Matching is accomplished via stemming, lemmatization, Wordnet synonymy and abbreviation expansion, and each group  $G_k$  is closed under transitivity. We expect that tokens from large groups, i.e., occurrences in multiple sentences or repeated occurrences in a single sentence, will be more likely to appear in the output. In the

following section, we design features over supporting tokens so that the learning algorithm can encourage or discourage their occurrence following the patterns seen in the training corpus.

#### 3.3.2 Redundancy constraints

While we expect largely positive weights on features for supporting tokens, this will also have the effect of encouraging of more than one token from the same group to occur in the output. In order to avoid this problem, we add a constraint for each group  $G_k \in \mathcal{G}$  that prevents tokens within a group from appearing more than once.

$$\sum_{i:t_i \in G_k} x_i \leq 1, \quad \forall G_k \in \mathcal{G} \quad (10)$$

### 3.4 Features

We now describe the features  $\phi$  over tokens and bigrams that guide inference for fusion instances.

- **Salience:** Fluent output fusions might require specific words to be preserved, highlighted or perhaps rejected. This can be expressed through features on token variables that indicate *a priori* salience, for which we consider patterns of part-of-speech (POS) tags and dependency arc labels obtained from input parses. Specifically, we define indicator features for POS sequences of length up to 2 that surround the token and the POS tag of the token's syntactic governor conjoined with the label. We also maintain features for whether tokens appear within parentheses and if they are part of a capitalized sequence of tokens (an approximation of named entity markup).
- **Fluency:** These features are intended to capture how the presence of a given bigram contributes to the overall fluency of a sentence. The bigram variables are scored with a feature expressing their log-likelihood under an LM. We also include features that indicate the sequence of POS tags and dependency labels corresponding to the tokens an bigram variable covers.

- **Fidelity:** One might reasonably expect that many bigrams in the input sentences will appear unchanged in the output fusion. We therefore propose boolean features that indicate whether a bigram was seen in the input.
- **Pseudo-normalization:** A major drawback of using linear models for generation problems is an inability to employ output sentence length normalization when scoring structures. Word penalty features are used for this purpose following their use in machine translation (MT) systems. These features are simply set to 1 for every token and bigram and their parameters are intended to balance out biases in output length that are induced by other features.
- **Support:** We note the amount of support—repetitions across input sentences—for nouns, verbs, adjectives and adverbs, as described in §3.3. We define features that count the number of repetitions for each of these tokens, and conjoin this with the POS class of each token. We also include binary variants of these features that indicate whether a token has support across 2, 3 or 4 input sentences. The constraint in (10) prevents these features from encouraging redundancy in the output.

Each scale-dependent feature is recorded absolutely as well as normalized by the average length of an input sentence. This was done in order to encourage the model to be robust to variation in sentence length during training.

### 3.5 Learning

The structured perceptron (Collins, 2002) was used in our experiments to recover good parameter settings  $w^*$  for the above features from training corpora. We used a fixed learning rate, averaged parameters over all iterations, and tracked performance in each epoch against a held-out development corpus. Following Martins et al. (2009), inference was sped up during training by only solving an LP relaxation of the fusion ILP.

## 4 Experiments

In order to evaluate our proposed fusion approach, we ran experiments over the corpus described in §2. For ease of reproducibility, we did not split the corpus randomly, rather, the 593 instances from

the DUC evaluations covering the years 2005–2007 were chosen as a testing corpus, while the 1265 instances from the TAC evaluations over 2008–2011 were used as a training corpus. This yields an approximate 70/30 train-test split with near-identical proportions of 2-way, 3-way and 4-way fusions. In addition, we used 10% of the training section (all from 2011) as a development corpus in order to tune the features.

Dependency parses for features were generated using the Stanford parser<sup>7</sup> and LMs were constructed from the Gigaword corpus. All ILPs were solved using Gurobi.<sup>8</sup> All possible token orderings were permitted for fusion inference with the exception of those that flipped the order of two tokens from the same input sentence, which we assumed to be highly unlikely.

### 4.1 Baselines

The lack of a standard corpus and domain makes comparisons against previous systems difficult. Indeed, we propose that the pyramid fusion corpus described here may be well suited for comparing fusion systems in the future.<sup>9</sup>

We therefore use two baselines for this evaluation. First, we consider a compression baseline that is a variant of the system under study but without the fusion-specific modifications, i.e., the support features and the redundancy constraint from (10). This is not a strong baseline—we do not expect it to outperform our system for this task—but it serves as a useful measure of how linearization performs in the absence of content selection.

Our second baseline uses an identical system to the first but operates on different input data—the SCU *contributors* for each instance instead of the full source sentences. These are human-selected text spans that realize the SCU as defined in the pyramid evaluation guidelines and therefore approximate gold content selection. One-third of the instances in the corpus (659 instances) have SCUs that are exact string matches of one of the contributors;<sup>10</sup> the corresponding count for SCU-matching source sentences is less than half (300 instances).

<sup>7</sup><http://nlp.stanford.edu/software/>

<sup>8</sup><http://www.gurobi.com>

<sup>9</sup>We hope to eventually distribute the extracted corpus directly but interested researchers can currently retrieve the raw data from NIST and reconstruct it from our guidelines in §2.

<sup>10</sup>We chose to leave these contributors in the corpus in order to more accurately model the decisions of human annotators who were generating the fusions.

Configuration	Input	n-grams F <sub>1</sub> %				Content words			Syntactic rels F <sub>1</sub> %	
		<i>n</i> = 1	2	3	4	P%	R%	F <sub>1</sub> %	Stanford	RASP
Compression	Sources	27.08	14.97	8.64	4.85	40.05	28.20	30.17	14.19	12.71
	Contribs	36.38	22.43	14.72 <sup>†</sup>	10.04 <sup>†</sup>	<b>55.27<sup>†</sup></b>	36.79	39.95	<b>22.81<sup>†</sup></b>	20.24 <sup>†</sup>
+ Support	Sources	<b>40.46<sup>†</sup></b>	<b>24.92<sup>†</sup></b>	<b>16.33<sup>†</sup></b>	<b>11.00<sup>†</sup></b>	49.01	<b>45.09<sup>†</sup></b>	<b>44.42<sup>†</sup></b>	<b>22.81<sup>†</sup></b>	<b>21.25<sup>†</sup></b>

Table 2: Experimental results under various quality metrics (see text for descriptions). Boldfaced entries in each column indicate statistically significant differences ( $p < 0.05$ ) over other entries under Wilcoxon’s signed rank test while <sup>†</sup> indicates the same under the paired t-test.

## 4.2 Evaluation Metrics

Sentence fusion is notoriously hard to evaluate (Daumé III and Marcu, 2004) and previous work tends to rely on human evaluations with Likert scales. However, we choose to follow work in machine translation and, more recently, sentence compression (Napoles et al., 2011; Thadani and McKeown, 2013) in moving towards transparent automated metrics for fusion quality in order to engender more repeatable evaluations of fusion systems. As our test corpus is larger than most previously-studied fusion corpora in their entirety, statistical measures of text quality are preferable.

Our basic evaluation metric is n-gram F<sub>1</sub>, used in numerous tasks and evaluation scenarios; we consider all  $1 \leq n \leq 4$ . In addition, since n-gram metrics do not distinguish between content words and function words, we also include an evaluation metric that observes the precision, recall and F-measure of only nouns and verbs as a proxy for the informativeness of a given fusion.

In addition to the direct measures discussed above, we consider syntactic metrics that act as surrogates for grammaticality. Napoles et al. (2011) indicates that F<sub>1</sub> metrics over syntactic relations such as those produced by the RASP parser (Briscoe et al., 2006) correlate significantly with human judgments of compression quality; we expect that the same holds for our fusion scenario. Output fusions were therefore parsed with RASP as well as the Stanford dependency parser and their resulting dependency graphs were compared to those of the gold fusions.

## 4.3 Results

Table 2 summarizes the results from the fusion experiments. We first observe that the proposed fusion system as well as the contributor+compression baseline outperform the source+compression baseline significantly on all metrics evaluated. We also observe a significant

gain for the fusion system over all baselines for F<sub>1</sub> over unigrams and bigrams, vindicating the proposed content-selection extensions to the baseline compression approach. Results for trigrams and 4-grams are statistically indistinguishable under the paired t-test, indicating that the proposed system is at least competitive with the ‘cheating’ baseline.

Turning to the content-word metrics, we see that the primary contribution of the discriminative joint approach is in enhancing the recall of meaning-bearing words. The gain in recall is larger than the loss in precision against the contributor+compression baseline, leading to a significant improvement in content word F<sub>1</sub>.

Finally, the results from the syntactic measures of fluency are less clear. The proposed fusion system outperforms the strong baseline on RASP F<sub>1</sub> but the gain is only statistically significant under Wilcoxon’s signed rank test. Both systems significantly outperform the weaker baseline.

Table 3 contains an example of system output illustrating the quirks of the different systems. We note that the results are often noisy in all scenarios and that the supported tokens do not entirely override the LM. For example, ‘ABC’ appears in only one of the input sentences in the second example from Table 3 but is seen in multiple system fusions, likely due to the influence of an LM trained on newswire text.

## 5 Discussion & Future Work

While the focus of this paper is on linearization, we also considered expanding the objective from (2) to include syntactic structures as presented in Thadani and McKeown (2013); however, initial results were not promising. We hypothesize that this is partly to the vulnerability of such representations to parse errors—also noted in Filipova (2010)—and partly to the severe independence assumptions involved in arc-factored dependency representations which are exacerbated when

Input 1	<b>Elián returned to Cuba on June 28, 2000.</b>
Input 2	After a final appeal by the Miami relatives was denied and the court order blocking his return expired, <b>Elián returned with his father to Cuba on June 28, 2000.</b>
Input 3	<b>On June 28</b> , the Supreme Court rejected a final appeal; <b>Elián returned home to Cuba</b> , was celebrated in the media and returned to his home and schooling.
Gold	Elián returned with his father to Cuba on June 28, 2000
Comp	Elián returned to Cuba on June returned with his father rejected a final appeal
Contribs	Elián returned to home to Cuba
+Support	Elián returned to Cuba on June 28
Input 1	<b>Jennings, who quit smoking several years ago, will undergo chemotherapy in New York.</b>
Input 2	ABC announced that Jennings would continue to anchor the news <b>during chemotherapy treatment</b> , but he was unable to do so.
Input 3	Peter Jennings hoarsely announced he had lung cancer on April 5, 2005 and <b>would begin outpatient chemotherapy in New York.</b>
Gold	Jennings will undergo chemotherapy in New York
Comp	ABC announced that 2005
Contribs	would begin outpatient chemotherapy chemotherapy treatment
+Support	ABC announced that Jennings would undergo chemotherapy in New York

Table 3: Examples of system outputs for instances from the corpus. Contributors are indicated by boldfaced text spans.

working with multiple input sentences. We are currently working on extending this approach to produce richer formulations of syntax that will be more appropriate for this task.

## 6 Related Work

Sentence fusion is the general label applied to tasks which take multiple sentences as input to produce a single output sentence. Barzilay & McKeown (Barzilay et al., 1999; Barzilay and McKeown, 2005) first introduced fusion in the context of multidocument summarization as a way to better capture the information in a cluster of related sentences than just using the centroid. The fusion task has since expanded to include other forms of sentence combination, such as the merging of overlapping sentences in a multidocument context (Marsi and Krahmer, 2005; Krahmer et al., 2008; Filippova and Strube, 2008b) and the combination of two (usually contiguous) sentences from a single document (Daumé III and Marcu, 2004; Elsner and Santhanam, 2011). Variations on the fusion task include the set-theoretic notions of *intersection* and *union* (Marsi and Krah-

mer, 2005; McKeown et al., 2010), which forego the problem of identifying relevance and are thus less dependent on context. Query-based versions of these tasks have been studied by Krahmer et al. (2008) and have produced better human agreement in annotation experiments than generic sentence fusion (Daumé III and Marcu, 2004). McKeown et al. (2010) produced an annotated fusion corpus which was employed in experiments on decoding for sentence intersection (Thadani and McKeown, 2011). While most work in the area has covered pairwise sentence combination, recent work by Filippova (2010) has also addressed fusion—referred to as multi-sentence compression—within a cluster of sentences.

## 7 Conclusion

We have presented a new corpus for sentence fusion which is built from readily-available data used for summarization evaluation. To our knowledge, this is the largest corpus of fusion data studied to date. In addition, we proposed a supervised discriminative approach for sentence fusion that jointly selects content from the input and recovers a linearization without an intermediate representation. Our system uses a flexible integer linear programming formulation for generating acyclic paths in token graphs, generalizing a state-of-the-art sentence compression approach to multiple sentences and a supervised setting which permits rich, linguistically-motivated features that factor over tokens and n-grams. We demonstrate that this approach leads to significant performance gains over a baseline compression system as well as comparable performance to an approach which directly leverages human content selection.

## Acknowledgments

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

## References

- Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, September.
- Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of ACL*, pages 550–557.
- Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of ACL-HLT*, pages 481–490.
- Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the ACL-COLING Interactive Presentation Sessions*.
- James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: an integer linear programming approach. *Journal for Artificial Intelligence Research*, 31:399–429, March.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models. In *Proceedings of EMNLP*, pages 1–8.
- Hal Daumé III and Daniel Marcu. 2004. Generic sentence fusion is an ill-defined summarization task. In *Proceedings of the ACL Text Summarization Branches Out Workshop*, pages 96–103.
- Micha Elsner and Deepak Santhanam. 2011. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 54–63.
- Katja Filippova and Michael Strube. 2008a. Dependency tree based sentence compression. In *Proceedings of INLG*, pages 25–32.
- Katja Filippova and Michael Strube. 2008b. Sentence fusion via dependency graph compression. In *Proceedings of EMNLP*, pages 177–185.
- Katja Filippova. 2010. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of COLING*, pages 322–330.
- Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *Proceedings of NAACL*, pages 178–185.
- Emiel Krahmer, Erwin Marsi, and Paul van Pelt. 2008. Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In *Proceedings of ACL-HLT*, pages 193–196.
- Thomas L. Magnanti and Laurence A. Wolsey. 1994. Optimal trees. In *Technical Report 290-94*, Massachusetts Institute of Technology, Operations Research Center.
- Erwin Marsi and Emiel Krahmer. 2005. Explorations in sentence fusion. In *Proceedings of the European Workshop on Natural Language Generation*, pages 109–117.
- André F. T. Martins, Noah A. Smith, and Eric P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL-IJCNLP*, pages 342–350.
- Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*, pages 297–304.
- Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *Proceedings of HLT-NAACL*, pages 317–320.
- Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97.
- 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.
- Kapil Thadani and Kathleen McKeown. 2011. Towards strict sentence intersection: decoding and evaluation strategies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 43–53.
- Kapil Thadani and Kathleen McKeown. 2013. Sentence compression with joint structural inference. In *Proceedings of CoNLL*.

# Detecting and Correcting Learner Korean Particle Omission Errors

**Ross Israel**  
Indiana University  
raisrael@indiana.edu

**Markus Dickinson**  
Indiana University  
md7@indiana.edu

**Sun-Hee Lee**  
Wellesley College  
slee6@wellesley.edu

## Abstract

We detect errors in Korean post-positional particle usage, focusing on optimizing omission detection, as omissions are the single-biggest factor in particle errors for learners of Korean. We also develop a system for predicting the correct choice of a particle. For omission detection, we model the task largely on English grammatical error detection, but employ Korean-specific features and filters; likewise, output analysis and the omission correction system illustrate how unique properties of Korean, such as the distinct types of particles used, need to be accounted for in adapting the system, thereby moving the field one step closer to robust multi-lingual methods.

## 1 Introduction

Grammatical error detection is useful to produce an improved final document for writing assistance, provide feedback to language learners, provide features for automatic essay scoring, and post-edit machine translation output (see references in Chodorow et al., 2012, sec. 2). Within this growing field, most of the work has focused on English, but there has been a small community of researchers working on other languages. We continue this trend by advancing the state-of-the-art in detecting errors in Korean particle usage.

Expanding to other languages and language families obviously presents new challenges, such as being able to handle word segmentation and greater morphological complexity (e.g., Basque (de Ilarraza et al., 2008), Korean (Lee et al., 2012), Hungarian (Dickinson and Ledbetter, 2012), Japanese (Mizumoto et al., 2011)); greater varieties of word order (Czech (Hana et al., 2010),

German (Boyd, 2012)); case ending errors (Czech, German, Hungarian); differing definitions of function words (Korean, Japanese, Basque); and so forth. Investing in methods which apply across languages will make techniques more robust and applicable for even more languages.

An additional challenge for many of these languages is the lack of resources. Much previous work on detecting errors in Korean, for example, focused less on techniques and more on acquiring training data (Dickinson et al., 2011) and evaluation data (Lee et al., 2012). We thus desire techniques that work using smaller and/or unannotated data sets that may be less reliable than some of the corpora for better-resourced languages.

We focus on detecting errors in the presence or absence of Korean postpositional particles. Korean is a Less Commonly Taught Language (LCTL) needing proficient speakers and more pedagogical research (see Dickinson et al., 2008, sec. 2), making computational tools for Korean language learning important. Particles are used to mark properties akin to prepositions and also to case markers, as discussed in section 2. This makes our task applicable to similar languages like Japanese and more generally to agglutinative languages like Basque, Hungarian, and Turkish, as discussed in section 3 on related work. Particles are our focus because of the high prevalence of particle errors in learner data, accounting for 20–30% of learner errors (section 2).

One of the most frequent errors relating to particles is not using them when required (section 4); thus, simply detecting whether a particle is necessary can pinpoint nearly half the particle errors language learners make. In the interest of extending methods to new languages, we develop an omission error detection system rooted in work on



English preposition error detection (section 5), accounting for Korean-specific properties in the features and filtering of results (section 6). We then see the impact of such error detection on predicting the specific omitted particles (section 7).

We make the following contributions in this paper: 1) We present a functional Korean particle omission error detection system, adapted from previous English preposition work but tailored towards Korean in its morpheme-based approach and its novel features. 2) We outline system mistakes, highlighting unique properties of Korean, and point towards how to fix them. 3) We provide an error correction system—incorporating new discourse-based features and optimized separately from the first-stage classifier—which corrects a high percentage of omission errors. In so doing, we also discover that accounting for distinctions in types of Korean particles opens the door to further improvements. The overall lesson is that work from English can be adapted, but only if incorporating the nuances of the new language.

## 2 Korean Particles

Korean postpositional particles are units that appear after a nominal to indicate different linguistic functions, including grammatical functions, e.g., subject and object; semantic roles; and discourse functions. In (1), for instance, *가* (*ka*) marks the subject (function) and agent (semantic role).<sup>1</sup>

- (1) 그래서 제가 한국말을 열심히 배우고  
 thus I-SBJ Korean-OBJ hard learn  
 싶어요  
 want  
 ‘Thus, I really want to learn Korean’

Similar to English prepositions, particles can also have modifier functions, adding meanings of time, location, instrument, possession, and so forth. For further discussion of Korean particles, see, e.g., chapter 3 of Yeon and Brown (2011).

**Learner Errors** Particle errors are very frequent for Korean language learners, accounting for 28% of beginner errors in one corpus study (Ko et al., 2004). In (2a), for instance, a learner omitted a subject particle after the word *것* (*kes*, ‘thing’). The error has been corrected in (2b).

- (2) a. 각 곳에 여러 좋은 것 있어요  
 each place-AT many good thing exist

- b. 각 곳에 여러 좋은 것이  
 each place-AT many good thing-SBJ  
 있어요  
 exist  
 ‘There are many good things’

## 3 Related Work

While there is much related work on detecting preposition and article errors in English, e.g., the 2012 Helping Our Own (HOO) shared task (Dale et al., 2012), we will focus here on work on detecting errors in functional items in agglutinative languages (Korean, Japanese, Basque), as we most directly build from this. Roughly, agglutinative languages here are ones which “glue” syntactic categories, in the form of affixes, onto a word.

For Korean particle error detection, Dickinson and Lee (2009) train two parser models, one with particles included and one without, to compare mismatches. Their main purpose is to adapt tree-bank annotation to be more particle-aware, and they did not evaluate on real learner data.

We build more directly from Dickinson et al. (2011), who build web corpora of Korean in order to train machine learning models for particle prediction. They obtain 81.6% accuracy for particle presence. While the work is similar, comparing the current work to their results is problematic for a number of reasons. First, the work in Dickinson et al. (2011) was very preliminary, focusing on acquiring training data, and did not examine different levels of learners. Also, they used a different learner corpus with different annotation guidelines (see comparison in Lee et al., 2012), along with training data that was specifically tailored for the domains in the test corpus. Finally, for particle presence, they focus on overall system accuracy, rather than error detection, making direct comparison of results difficult (cf. Chodorow et al., 2012).

There has been more work in the comparable language of Japanese, which we review briefly. To begin with, Oyama (2010) uses a basic SVM model trained on well-formed Japanese to detect particle errors, focusing on eight different case particles and finding that the particle frequency distribution in the training corpus affects accuracy, ultimately evaluating on 200 learner particle instances of a single particle (*wo*).

Mizumoto et al. (2011) use statistical machine translation (SMT) techniques to detect and correct all errors within Japanese, using a “parallel” cor-

<sup>1</sup>Examples come from the learner corpus; see section 4.2.

pus of ill-formed and correctly-formed Japanese, based on correction logs from a collaborative language learning website. Our paradigm is much different, basing our method only on a correct model of the target language, given a relative lack of corrected data available in Korean and other lesser-resourced languages. We are, however, able to use some correction logs for building confusion sets (section 7.1). Imamura et al. (2012) correct Japanese particle errors using an approach similar to SMT ones, relying on a corpus of generated errors to learn a model of alignment to correct forms. We could explore generated errors in the future, but rely only on a model of correct Korean here.

Suzuki and Toutanova (2006) predict case markers in Japanese for an MT system, basing their techniques on semantic role labeling. They predict 18 case particles, a subset of all Japanese particles. They use a two-stage classifier, first identifying whether case is needed and then assigning the particular case ending, training the second classifier only on instances where a case marker was required. This breakdown and parts of their feature sets are similar to ours, but: a) they use (gold standard) parse features and treat the problem as one of predicting markers for *phrases*; and b) they correct machine errors, while we correct learner errors, allowing us to investigate methods such as using learner-based filters.

Turning to Basque, de Ilarraza et al. (2008) detect errors in five complex postpositions, where the postposition itself has a suffix, by developing 30 constraint grammar rules which use morphological, syntactic, and semantic information. While the rule-based system can work well, we pursue a strategy which incorporates different types of linguistic information through contextual features.

## 4 Data

### 4.1 Training Data: Collecting Web Data

In order to control the data for domain specificity, we follow the recommendations laid out in Dickinson et al. (2010) and extended in Dickinson et al. (2011). Namely, we use data collected from the web using search terms based on topics likely to be discussed in a learner corpus, in order to find semantically-relevant instances. This data is passed through an encoding filter to ensure that at least 90% of any document retrieved is written using Hangul (the Korean writing system). The resultant corpus is over 23 million words.

### 4.2 Testing Data: A Learner Korean Corpus

For testing data, we use a corpus of learner Korean (Lee et al., 2012, 2013) featuring 100 error-annotated essays from learners evenly split into four different categories: beginning (B) vs. intermediate (I) learners, and foreign (F) vs. heritage (H) learners, where *heritage* refers to learners who had Korean spoken at home.<sup>2</sup> We split the corpus into development and test sets by taking  $\approx 20\%$  of each subcorpus for development, and using the rest as testing. Table 1 gives the numbers of sentences, tokens, nouns, particles, total errors, and omission errors in the development and testing sets.

	Sen.	Tok.	Noun	Part.	Err.	Om.
Dev	331	2673	955	849	103	51
Test	1079	8987	3266	2872	430	234
All	1410	11660	4221	3721	533	285

Table 1: Annotated corpus statistics (*sentences, tokens, nouns, particles, errors, omissions*)

Particle errors are marked as omissions, insertions (omissions), or substitutions, in a multi-layered framework. Spacing and spelling errors are corrected before the target form and correct segmentation are marked, segmentation being necessary since nouns and particles are written as a single orthographic unit. For our experiments, we use the correctly-spelled layer, mitigating the effect of spelling errors for testing an error detection system, as done for English (e.g., Tetreault and Chodorow, 2008; Chodorow et al., 2007).

All particles (erroneous or correct) are labeled as to their function (e.g., locative), allowing us to group particles into categories, to see how classifier performance differs. Figure 1 provides the four groups we consider (cf. tables 5 and 6). Additionally, some nominals require multiple particles in sequence (*Seq.*), and some of the annotations allow for particles from more than one category as a correct answer, i.e., a set of correct answers (*Set*).

### 4.3 Learner Error Analysis

Lee et al. (2009) annotate another corpus of learner Korean, divided using the same four-way split among learner level and type as the corpus described in section 4.2. We examine this corpus

<sup>2</sup>The corpus will be publicly released at: <http://cl.indiana.edu/~kolla/>.

Category	Example Functions
Structural Case	subject, object, genitive
Inherent Case	time, location, goal, etc.
Auxiliary	auxiliary, topic
Conjunction	conjunction

Figure 1: Particle Categories

to get a sense of the types of errors that learners of Korean make in essays. In this corpus, omission errors, i.e., instances where the learner has mistakenly omitted a particle, make up the biggest proportion of the errors (47.6%). The next most common are replacement errors, where the learner has used the wrong particle (44.6%). Commission errors—using a particle where none is necessary—make up the remainder of the errors (7.8%).

## 5 Approach

Particles have a range of functions, including case marking and preposition-like functions, but, since they are a closed class of functional elements, we can adapt techniques from English for other closed class functional items, namely prepositions and articles, to detect errors in usage.

We view the task of detecting and correcting errors as two steps (cf. Gamon et al., 2008). The first step is a binary choice that only involves determining whether or not a particle is required, a so-called *presence* (yes/no) classifier. The second classifier, the particle *choice* classifier, attempts to guess the best particle, once it has been established that a particle is needed. We actually treat the first step as a particle omission detection system because the expected rate of errors of commission is so low, and thus we specify that the classifier cannot reject a particle that is already present. Commission errors may require their own system.

We utilize the omission classifier as it nicely performs two functions. First, because it posits instances requiring particles, it also filters out instances that do not need a particle to be grammatical. Thus, the particle choice classifier does not need to include *NULL* as a possible class, cutting down on training size and complexity. Secondly, many errors can be found at this stage, as a lot of errors stem from learners omitting necessary particles (see section 4.3). Nearly half of the learner errors could be detected with an accurate omission particle detection system at this step. Thus, this classifier can provide useful feedback to learners,

especially higher-level ones who may know the correct particle once its omission is highlighted.

## 6 Particle Omission Error Detection

We describe the particle presence classifier here, treating it as a task of particle omission detection. Any particle a learner uses is passed on, while we posit where a particle should have been used.

### 6.1 CRF Classifier

Conditional Random Fields (CRFs) have been utilized in a variety of NLP tasks in the last few years, and have been used recently for learner error detection tasks, especially those which can be seen as sequence labeling tasks (e.g., Israel et al., 2012; Tajiri et al., 2012; Imamura et al., 2012). We use the comma error detection work in Israel et al. (2012) as a basis, and employ CRF++<sup>3</sup> to set up a binary classifier at this step based on 1.5 million instances from our web corpus. Here we consider all nominals, as annotated in the corpus, as possible candidates for particle insertion. When we derive features based on POS tags (section 6.2), however, we rely on an automatic POS tagger.

### 6.2 Features

The feature set for particle omission detection is mainly composed of words and POS tags in the surrounding context, where tags are derived from a POS tagger (Han and Palmer, 2004). We use a five-word sliding window, processing each token in the document, although only nominals are possible candidates for particle insertion. The five-word window includes the target word and two words on either side for context; the feature set, with examples, is given in table 2.

We break all words into their root and a string of affixes, each with its own POS tag (or tags, for multiple affixes) to better handle the morphological complexity of Korean and avoid sparsity issues. Particles are removed when extracting affixes, so as not to include what we are trying to guess. For the text and POS of the root, we use unigram, bigram, and trigram features, as shown in the table; for the affixes, we use only unigrams. We also have a feature (*combo*) for each root that combines the text and POS into a single string.

In addition to these adjacency-based features, we also encode the previous and following nouns

<sup>3</sup><http://crfpp.googlecode.com/svn/trunk/doc/index.html>

position	Unigrams					Next		Prev		# of nouns passed	# of nouns remain						
	text		POS		combo	Noun	Pred	Noun	Pred								
	Root	Affix	Root	Affix													
Target	것	NONE	NNX	NONE	것_NNX	NONE	있	곳	좋	1	0						
Word <sub>-1</sub>	좋	은	VJ	EAN	좋_VJ	NONE	NONE	것	좋	1	1						
Word <sub>-2</sub>	여러	NONE	DAN	NONE	여러_DAN	것	좋	곳	NONE	1	1						
Word <sub>+1</sub>	있	어요	VJ	EFN	있_VJ	NONE	NONE	것	좋	2	0						
Word <sub>+2</sub>	.	NONE	SFN	NONE	..SFN	NONE	NONE	것	있	2	0						
Bigrams - text					Bigrams - POS												
W <sub>-2</sub> +W <sub>-1</sub>		W <sub>-1</sub> +T		T+W <sub>+1</sub>		W <sub>+1</sub> +W <sub>+2</sub>		W <sub>-2</sub> +W <sub>-1</sub>		W <sub>-1</sub> +T		T+W <sub>+1</sub>		W <sub>+1</sub> +W <sub>+2</sub>			
여러+좋		좋+것		것+있		있+.		DAN+VJ		VJ+NNX		NNX+VJ		VJ+SFN			
Trigrams - text					Trigrams - POS												
W <sub>-2</sub> +W <sub>-1</sub> +T			W <sub>-1</sub> +T+W <sub>+1</sub>			T+W <sub>+1</sub> +W <sub>+2</sub>			W <sub>-2</sub> +W <sub>-1</sub> +T			W <sub>-1</sub> +T+W <sub>+1</sub>			T+W <sub>+1</sub> +W <sub>+2</sub>		
여러+좋+것			좋+것+있			것+있+.			DAN+VJ+NNX			VJ+NNX+VJ			NNX+VJ+SFN		

Table 2: Features and examples for *것* in (2b) - '각 곳에 여러 좋은 것이 있어요'

and predicates, to approximate syntactic parent features. The *predicates* can be verbs, adjectives that function like verbs in Korean, and auxiliary verbs. Finally, we use two features to encode the amount of nouns that have already occurred in the sentence, as well as how many still remain. The usage of topic particles, for instance, relies in part on knowing where in the sentence a noun occurs, with respect to other nouns.

### 6.3 Filtering

Because learners are more often correct than erroneous in their usage of particles, we want to ensure that the output of classifier does not predict errors in too many instances. To this end, we have built a filter into the classifier. For these errors of omission, we check how confident the classifier is in its answer and only posit omission errors if the classifier's confidence is above a certain threshold. Tuning on the development corpus (section 6.4), we tried a variety of thresholds, in a hill-climbing approach, and found 85% to be the best.

### 6.4 Results

For all results in this paper, we follow the recommendations from Chodorow et al. (2012). We evaluate by comparing the writer, annotator, and system's answer for each instance; true positives (TP), for example, are cases where the annotator (gold standard) and system agree, but the writer (learner) disagrees. In our case, positives are cases where the system posits a particle while the learner did not. We count only instances of nominals without particles in the writer's data, as these are the only ones which could have omission errors. Along with precision (P), recall (R), and an F-score ( $F_{0.5}$ ), we provide the number of errors ( $n$ ),

true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), for the sake of clarity and future comparison. As a baseline, we use the majority class, i.e., guessing a particle for every nominal in the corpus.

Table 3 provides the results for particle omission detection on our development corpus. Here we present the baseline, the results based only on the classifier's decision (no filter), and the results for the best filter. We use precision-weighted  $F_{0.5}$  rather than the traditional  $F_1$  because precision is more important than recall for most error detection applications. As the 85% threshold results in the best  $F_{0.5}$ , we use this system on the test data.

Table 4 provides the results for particle omission detection broken down by subcorpus. The FB (foreign beginner) subcorpus has the worst performance, most likely due to their language being most distant from the well-formed Korean of the training corpus, as well as the most distant from the development set. Overall, however, the system has a solid 84.9% precision on all test subcorpora.

### 6.5 Analysis

In looking over some FPs, i.e., cases where the system predicted a particle not in the gold standard, we discovered that some of these cases involved the optionality of particles. For example, in (3), the system posits a particle after *사람들* (*salamtul*, 'people'). This is a case of a nominal being used in a genitive fashion, and so a genitive particle could be used here, but it is not required. In some sense, the system rightly points to particle usage being *licensed* in this setting. However, the corpus annotation only marks particles that are *necessary* for grammaticality (Lee et al., 2013). Fully teasing apart particle licensing from particle

	<i>n</i>	TN	TP	FP	FN	Precision	Recall	$F_{0.5}$
Dev baseline	51	0	51	98	0	34.23	100.00	39.41
Test baseline	233	0	233	391	0	37.34	100.00	42.69
Dev no filter	51	64	43	34	8	55.84	84.31	59.89
Dev 85% filter	51	90	27	8	24	77.14	52.94	70.68
Test 85% filter	233	373	101	18	132	84.87	43.35	71.23

Table 3: Particle omission error detection results

	<i>n</i>	TN	TP	FP	FN	Precision	Recall	$F_{0.5}$
FI	66	153	33	6	33	84.62	50.00	74.32
FB	51	45	18	5	33	78.26	35.29	62.94
HB	53	46	20	4	33	83.33	37.74	67.11
HI	63	129	30	3	33	90.91	47.62	76.92

Table 4: Particle omission error detection results by learner type (test data)

requirement requires more thorough discussion of when particle dropping is permitted.

- (3) 특히 외국 사람들 눈에는 더욱  
particularly foreign **people** eye more  
그렇습니다.  
is-so.

‘In particular, it is thus for the eyes of foreign people.’

Other cases do not license particles, but the nominals still have particle-like functions. In (4), for instance, the nominal phrase 이 때 (*i tay*, ‘this time’) carries a temporal meaning—much like that conveyed in the temporal particle 에 (*ey*), but no particle is allowed here, because the function is more like an adverb (cf. *today* in English).

- (4) 이 때 너무 감정에 치우치지  
**this time** too feeling-at give-way-to  
않도록 주의하여야 해.  
don’t pay-attention-to must.

‘This time, you must pay attention to not giving way to feeling.’

Regarding false negatives, i.e., cases where we do not posit a particle when we should, one major problem we observe involves noun-verb and noun-noun sequences. If a learner views a noun and a following word like a compound, it conceals the fact that the noun requires a particle. For instance, in (5) (learner-omitted particles in curly brackets), the word 성격 (*sengkyek*, ‘personality’) needs a subject particle, but it forms a compound with 좋 (*choh*, ‘good’), obscuring the noun’s role.

- (5) 성격{이} 좋은 아{가}  
personality{-**SBJ**} good-REL kid{-**SBJ**}  
태어날 때 환경이 나쁘다면 ...  
born time environment-**SBJ** bad-if ...  
‘When a child who has good personality is born, if the environment is bad ...’

Another complication is the variability of particle requirements due to minor changes in the amount of information presented, for example, the addition of one prepositional phrase changing whether a particle is necessary or not. Combined with misclassifications resulting from segmentation errors from the POS tagger, it seems like the false negative set can be reduced with better linguistic preprocessing fed into the system.

## 7 Particle Choice for Omission Errors

Once we have established that there is a missing particle, the next step is to select the best particle to be placed in the given context. Thus, we send all instances classified as missing a particle to a second classifier that makes this selection.

### 7.1 Confusion Set for Particle Omission

The scope of the training data selected, i.e. what particles should be allowed to be guessed by the classifier, is a significant decision at this stage. There are hundreds of particles in the Korean language, but many of these are not used often, e.g., 9 particles cover 70% of particle use in a data set of thesis abstracts and 32 cover 95% in a study by Kang (2002). Thus, the training data should only include particles which can reasonably be ex-

pected to appear when the learner has omitted one. Utilizing similar methodology to Mizumoto et al. (2011), we build a confusion set from data collected from the language learning and social networking website, Lang-8.<sup>4</sup>

To build the set, we searched the user-edited versions of the essays for any word corrected by appending text resembling a particle. Due to the somewhat ambiguous nature of particles with respect to other morphemes and root endings, we cannot be certain that all of these edits are in fact particles, but can be confident that a majority are.

After compiling all possible insertion candidates, we prune the list by requiring a particle’s frequency to be at least 10% of the most frequent particle. For example, if  $\text{ㄱ}$  appears 100 times as the most frequently inserted particle, any particle appearing less than 10 times would be removed.

## 7.2 TiMBL

For this task, we use memory-based learning, namely TiMBL (Daelemans and van den Bosch, 2005). The nearest neighbor algorithm is desirable as training data is sparse, and there are a variety of possible classes to choose from. After filtering the web-corpus to only include instances based on the confusion set extracted from the Lang-8 data, we have 5.7 million instances for training.

## 7.3 Features

For the particle selection system, we build upon the particle omission detection features (cf. section 6): we use unigrams, bigrams, and trigrams of the words and POS tags, a combination word+POS unigram, the previous and following verbs and nouns, and the count of nouns passed and remaining in the sentence. We only use nominals as targets for instances, using a five-word window for context. Some of the  $n$ -gram features with high numbers of possible values are less helpful, and we remove them, namely the unigram features for the two words farthest from the target, as well as the bigrams that do not include the target.

We then extend this information by adding features, some of which provide discourse information. 1) Knowing if there is already a subject, object, or topic particle in the sentence often means that there should not be another of the same type used; thus, we add binary features encoding if any of these have occurred yet. 2) We also add binary

features relating to the usage of the target word in the previous sentence, encoding if the target was marked as the topic, subject, or object, or if it was in the previous sentence at all. 3) A numeric feature is used that tracks how far along we are in the sentence, based on the idea that certain particles, e.g. subjects, are more likely to occur earlier in the sentence, whereas others, e.g. objects, occur later. 4) Finally, we include the previous particle used by the learner, again because some particles are not likely to be reused in a sentence.

## 7.4 Results and Analysis

Here we present the results for the selection classifier in terms of the accuracy of the classifier on choosing the best particle for an instance already defined as erroneous. By the definition of this task—selecting the correct particle for an *error*—there are no FNs or TNs. Thus, recall is rather meaningless, and accuracy and precision reduce to the same metric ( $\frac{TP}{TP+FP}$ ). Additionally, as mentioned in section 4.2, the particles in the test corpus can be grouped into different categories, and we provide results broken down by category and sub-corpus. Instances that require a sequence of multiple particles to be correct (*Seq.*) are not currently handled, but we leave them in the results for clarity and completeness. FPs from the error detection step are also included, although the system clearly cannot select a correct particle for them.

Table 5 shows the performance of the selection classifier on the instances identified as omission errors by the binary classifier (i.e., TPs and FPs identified by the *pipeline*). Overall, this classifier selects the correct particle 52.9% (63/119) of the time in the test data when presented with instances from the previous classifier.

	Dev	Test	FI	FB	HB	HI
Str.	17/20	56/80	16/28	11/15	14/15	15/22
Inh.	1/2	5/8	1/2	1/1	1/2	2/3
Aux.	0/1	1/3	0/1	0/0	0/1	1/1
Cnj.	0/1	0/0	0/0	0/0	0/0	0/0
Set	0/2	1/4	0/1	0/1	1/1	0/1
Seq.	0/1	0/6	0/1	0/1	0/1	0/3
FPs	0/8	0/18	0/6	0/5	0/4	0/3
Total	18/35	63/119	17/39	12/23	16/24	18/33
%	51.4	52.9	43.6	52.2	66.7	54.5

Table 5: Results for particle selection on instances from binary omission classifier (*pipeline*)

Table 6 provides the results for testing on all instances with omission errors (based on the *gold*

<sup>4</sup><http://lang-8.com>

standard), i.e., including the FN instances from the binary omission classifier mistakenly marked as correct, but not FPs. For all corpora combined, the classifier selects the best particle 58.4% (136/233) of the time in the test data. The overall accuracy gleaned from Tables 5 and 6 is encouraging as we move forward, as it means that the classifier performs reasonably well on cases where it has a chance of selecting the best particle in both the *pipeline* and *gold* experimental environments.

	Dev	Test	FI	FB	HB	HI
Str.	23/29	112/164	34/51	17/31	29/36	32/46
Inh.	3/8	11/30	1/7	2/7	3/6	5/10
Aux.	3/6	10/16	1/3	2/4	5/7	2/2
Cnj.	0/1	0/0	0/0	0/0	0/0	0/0
Set	0/3	3/7	1/2	0/1	2/2	0/2
Seq.	0/4	0/16	0/3	0/8	0/2	0/3
Total	29/51	136/233	37/66	21/51	39/53	39/63
%	56.9	58.4	56.1	41.1	73.6	61.9

Table 6: Results for particle selection on all instances of particle omission (*gold*)

### 7.5 Further Restricting the Task

It is clear from the number of errors for each particle category that structural particles are the type most often omitted by learners of Korean, accounting for 68% (193/284) of omission errors in all subcorpora combined. Based on this finding, we ran a set of experiments in which we trained a classifier to only insert structural case particles.

	<i>pipeline</i>		<i>gold</i>	
	Dev	Test	Dev	Test
Str.	17/20	67/80	23/29	133/164
%	85	83.8	79.3	81.1
Total	17/35	67/119	23/51	133/233
%	48.6	56.3	45.1	57.1

Table 7: Results for particle selection using a structural case-only classifier

This classifier actually performs better than when restricting selection to the particles from the confusion set for the *pipeline* experiment setting (cf. Table 5, 56% > 52%), though there is a slight drop in performance as compared to the confusion set classifier in the *gold* experiments (cf. Table 6, 57% < 58%). In both cases, however, there are significant gains made when only examining structural particles; this classifier correctly identifies the best particle over 80% of the time in both the *pipeline* and *gold* test settings. These results show the potential in handling specific linguistic

types of particles in Korean differently.

## 8 Conclusion and Outlook

We have presented a system for detecting and correcting learner Korean particle omission errors. We used a two-stage pipeline utilizing CRFs to make a binary decision as to whether or not a nominal without a particle should be followed by a particle, followed by a memory-based learner to select the best particle in the case of an omission. The binary classifier performs with 85% precision and 44% recall in the testing data, for an  $F_{0.5}$ -score of 71%; these results could lead to a useful error detection tool for learners and/or teachers. The selection classifier is also fairly accurate, choosing the best particle close to 60% of the time to correct omission errors. These results compare favorably with English preposition and determiner error correction work (cf. Dale et al., 2012), though those results involve all error types, not just omissions.

Our experiments for the selection task using specific particle types indicate that constraining the set of particles for a given context helps greatly. We saw improvement in choice accuracy by using only structural case particles to train a classifier for selecting structural case. This encouraging result can help direct research moving forward. One could build a classifier to identify what category of particle is most likely for a given context after determining a particle is missing and before sending it to a final selection classifier.

Finally, as we improve the omission detection/correction pipeline, the next logical step for building a tool for more robust grammatical error detection is to take on errors of substitution and commission. The lessons learned here from particle choice, using a feature set that incorporates dialog-based features and constraining the set of particles that can be selected for a given context, should prove particularly useful for the substitution task. Just as we have seen that structural case particles are the most likely to be dropped, we may be able to find patterns for what types of particles can be substituted or over-used by learners. Confusion sets for the types of errors made by learners (cf., e.g., Rozovskaya and Roth, 2010) should be even more useful for substitution errors.

## References

- Adriane Boyd. 2012. *Detecting and Diagnosing Grammatical Errors for Beginning Learners of German: From Learner Corpus Annotation to Constraint Satisfaction Problems*. Ph.D. thesis, The Ohio State University.
- Martin Chodorow, Markus Dickinson, Ross Israel, and Joel Tetreault. 2012. Problems in evaluating grammatical error detection systems. In *Proceedings of COLING-12*.
- Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, pages 25–30. Prague.
- Walter Daelemans and Antal van den Bosch. 2005. *Memory-Based Language Processing*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, UK.
- Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62. Montréal.
- Arantza Díaz de Ilarraza, Koldo Gojenola, and Maite Oronoz. 2008. Detecting erroneous uses of complex postpositions in an agglutinative language. In *Proceedings of COLING-08*. Manchester.
- Markus Dickinson, Soojeong Eom, Yunkyung Kang, Chong Min Lee, and Rebecca Sachs. 2008. A balancing act: How can intelligent computer-generated feedback be provided in learner-to-learner interactions. *Computer Assisted Language Learning*, 21(5):369–382.
- Markus Dickinson, Ross Israel, and Sun-Hee Lee. 2010. Building a Korean web corpus for analyzing learner language. In *Proceedings of the 6th Workshop on the Web as Corpus (WAC-6)*. Los Angeles.
- Markus Dickinson, Ross Israel, and Sun-Hee Lee. 2011. Developing methodology for Korean particle error detection. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 81–86. Portland, OR.
- Markus Dickinson and Scott Ledbetter. 2012. Annotating errors in a Hungarian learner corpus. In *Proceedings of LREC 2012*. Istanbul.
- Markus Dickinson and Chong Min Lee. 2009. Modifying corpus annotation to support the analysis of learner language. *CALICO Journal*, 26(3).
- Michael Gamon, Jianfeng Gao, Chris Brockett, Alexander Klementiev, William Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for esl error correction. In *Proceedings of IJCNLP-08*. Hyderabad, India.
- Chung-Hye Han and Martha Palmer. 2004. A morphological tagger for korean: Statistical tagging combined with corpus-based morphological rule application. *Machine Translation*, 18(4):275–297.
- Jirka Hana, Alexandr Rosen, Svatava Škodová, and Barbora Štindlová. 2010. Error-tagged learner corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 11–19. Uppsala, Sweden.
- Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. Grammar error correction using pseudo-error sentences and domain adaptation. In *Proceedings of ACL-12 - Volume 2, ACL '12*, pages 388–392. Stroudsburg, PA, USA.
- Ross Israel, Joel Tetreault, and Martin Chodorow. 2012. Correcting comma errors in learner essays, and restoring commas in newswire text. In *Proceedings of NAACL-HLT 2012*.
- Seung-Shik Kang. 2002. *Korean Morpheme Analysis and Information Retrieval (in Korean)*. Hongrungs Publishing Company.
- S. Ko, M. Kim, J. Kim, S. Seo, H. Chung, and S. Han. 2004. *An analysis of Korean learner corpora and errors*. Hanguk Publishing Co.
- Sun-Hee Lee, Markus Dickinson, and Ross Israel. 2012. Developing learner corpus annotation for Korean particle errors. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 129–133. Jeju, Republic of Korea.
- Sun-Hee Lee, Markus Dickinson, and Ross Israel. 2013. Challenges in annotating korean particle errors. In *20 years of learner corpus research: looking back, moving ahead (LCR 2011)*.
- Sun-Hee Lee, Seok Bae Jang, and Jae-Young Song. 2009. Particle errors in an annotated Korean learner corpus - a comparative analysis of heritage learners & non-heritage learners-. In *Proceedings of Annual Conference of American Association of Teachers of Korean*. Seattle, USA.
- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of IJCNLP-12*, pages 147–155. Chiang Mai, Thailand.
- Hiroshi Oyama. 2010. Automatic error detection method for japanese particles. *Polyglossia*, 18.
- Alla Rozovskaya and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. In *Proceedings of EMNLP-10*, pages 961–970. Cambridge, MA.
- Hisami Suzuki and Kristina Toutanova. 2006. Learning to predict case markers in japanese. In *Proceedings of COLING-ACL-06*, pages 1049–1056. Sydney.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of ACL-12: Short Papers - Volume 2, ACL '12*, pages 198–202. Stroudsburg, PA, USA.
- Joel Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of COLING-08*. Manchester.
- Jaehoon Yeon and Lucien Brown. 2011. *Korean: A Comprehensive Grammar*. Routledge, New York.



# Automatic Identification of Learners' Language Background based on their Writing in Czech

**Katsiaryna Aharodnik<sup>1,2</sup>**

katiaaharodnik@gmail.com

**Marco Chang<sup>1</sup>**

changreynam1@mail.montclair.edu

**Anna Feldman<sup>1</sup>**

anna.feldman@montclair.edu

**Jirka Hana<sup>3</sup>**

jirka.hana@gmail.com

<sup>1</sup>Montclair State University, New Jersey, USA

<sup>2</sup>City University of New York, The Graduate Center, New York, USA

<sup>3</sup>Charles University, MFF, Czech Republic

## Abstract

The goal of this study is to investigate whether learners' written data in highly inflectional Czech can suggest a consistent set of clues for automatic identification of the learners' L1 background. For our experiments, we use texts written by learners of Czech, which have been automatically and manually annotated for errors. We define two classes of learners: speakers of Indo-European languages and speakers of non-Indo-European languages. We use an SVM classifier to perform the binary classification. We show that non-content based features perform well on highly inflectional data. In particular, features reflecting errors in orthography are the most useful, yielding about 89% precision and the same recall. A detailed discussion of the best performing features is provided.

## 1 Introduction

The role of a learner's native language (L1) in second language (L2) acquisition has been widely discussed in the theories of Second Language Acquisition (SLA) (Lado, 1957; Richards, 1971; Corder, 1975). The literature suggests that writers' spelling, grammar and lexicon in second languages are often influenced by patterns in their native language. However, the extent of the importance of L1 for acquiring L2 still cannot be determined exactly and remains a controversial topic of SLA research. Recently, the availability of learner corpora (e.g., Granger, 2003) has provided opportunities for verifying

SLA hypotheses. The previous literature suggests that the best performing features for native language identification are largely the features that rely on the content of the data, such as word n-grams, function words and character n-grams (Kochmar, 2011; Koppel et al., 2005; Tsur et al., 2007). This means that future applicability of these features is limited to corpus specific data. The primary goal of our work is to address this problem. We use only non-content based features, part-of-speech tags (POS) and error tags. Exploring these features is useful for corpora independent approaches to native language identification. Our secondary goal is to analyze the features that perform best for highly inflectional data. We approach binary classification as the beginning step in the development of a systematic tool for recognizing a specific L1 from morphologically complex L2 data. We use machine learning techniques to identify features contributing to the classification between Indo-European (IE) and non-Indo-European (NIE) L1 backgrounds of learners of L2 Czech. We employ Support Vector Machines (Joachims, 1999) to perform the classification. The results of the experiments show that the non-content based features, especially error tags, are the strongest indicators of the learners' language background.

## 2 Related Work

The task of native language identification has branched out from authorship attribution and profiling. For instance, Mosteller and Wallace (1964) have worked with the Federalist Papers to

identify the papers' authors. They looked at function words as their features. There is plenty of work addressing authorship profiling for data in languages other than English, for instance Dutch, Greek and Arabic (Van Halteren, 2004; Stamatos et al., 2001; Estival et al., 2007).

For automatic native language identification researchers have been exploiting learner corpora (Koppel et al., 2005; Wong and Dras, 2009; Tsur et al., 2007; Kochmar, 2011). Several SLA theoretical foundations have been taken as the basis for this task, for instance, the Contrastive Analysis Hypothesis (CAH) (Lado, 1957). CAH posits that difficulties in second language learning are derived from the differences between the source and target languages. It is expected that the more similar L1 and L2 are, the acquisition of L2 is more natural for a learner, and fewer mistakes are made, or positive transfer takes place. At the same time, learners have more difficulties in acquiring L2 if there are more differences between the source and target languages, which results in negative transfer, or errors. Richards (1971) addresses the nature of errors within the Error Analysis approach. He outlines interlingual and developmental types of errors. Developmental errors are common errors for any learner of a given L2, while interlingual errors are specific for each L1 or a group of L1s. Hence, interlingual errors should possess a discriminatory nature (Corder, 1975) and are of primary interest for the purpose of the native language identification.

In the search for empirical evidence, the researchers have looked at learners' errors and other idiosyncrasies in non-native writings as cues to predict a learner's native language and to conform to the above theoretical approaches as well as the phenomenon of language transfer in particular (Jarvis et al., 2012; Tsur et al., 2007). Koppel et al. (2005) look at 185 error types, including misspellings and syntactic errors as features. Besides errors, function words, character n-grams, and rare POS bigrams of non-standard English extracted from the Brown Corpus are used in the study. Koppel et al. (2005) experiment with essays from the International Corpus of Learner English in five source languages: Bulgarian, Czech, French, Russian and Spanish, and demonstrate that some types of errors are particularly useful for native language identification. Koppel et al. (2005) report slightly above 80% accuracy (with all features combined) compared to 20% baseline for 5-class classification. However, it is unclear from the study if the utili-

zation of error-based features would improve the performance with the same significance if taken on their own. We can only infer from the diagram in the paper that error features perform at slightly higher than 50% on their own and do not contribute significantly to the performance when combined with other features. Koppel et al. (2005) make valuable observations about function words and character n-grams as the most discriminative features.

Wong and Dras (2009) explore the contribution of three syntactic errors to the same task: subject-verb disagreement, noun-number disagreement and misuse of articles. The L1 backgrounds in the experiments are Bulgarian, Czech, French, Russian, Spanish, Chinese and Japanese. The accuracy obtained from classification based on these three features is 24.57% for multi-class classification. This, compared to the baseline of 14.29%, appears to be a significant improvement at the 95% confidence level. To achieve better results, the syntactic features are combined with function words, character n-grams and POS n-grams. The best accuracy is 73.71% using a combination of all the features. The results of this study demonstrate that the three syntactic errors do not contribute noticeably to classification if used without other features.

Tsur et al. (2007) investigate native language identification in the domain of phonology. Tsur et al. (2007) work with essays of Bulgarian, Czech, Russian, Spanish and French L1 backgrounds. The essays are taken from the International Corpus of Learner English. Tsur et al. (2007) suggest that learners' L1 has a strong effect on word choice in their L2 writings. The results of the classification, based only on character bi-grams, yield an accuracy of 66% in a 5-class task. The results demonstrate that the learners' choice of words when writing in a second language is influenced by the phonology of their native language suggesting evidence for language transfer (Tsur et al., 2007).

Kochmar (2011) explores the Cambridge Learner Corpus and provides a systematic error analysis for a number of two-class classification experiments. From her results, we can see that errors contribute to native language identification for learner English data. The highest result of 100% classification accuracy is achieved for misspelled character quad-grams for the Danish-Swedish group of languages. Besides specific L1s, she also looks at binary classification between language families, such as Romance and Germanic. The best result, 84% accuracy, for

this group is achieved for the combination of character tri-grams, POS n-grams and corpus derived error rates.

While the previous research widely exploits content based features, in our work we evaluate the usability of non-content based features and show that these features are reliable cues for native language identification. Moreover, all the above studies focus on learners' writings in English. In our work we investigate learner Czech data.

### 3 Experiments

#### 3.1 Corpus

We use the Czech as a Second Language (CzeSL) (Hana et al., 2010; Jelinek et al., 2012; Rosen et al., 2013) corpus, a newly developed learner corpus of Czech. Czech is a West Slavic language that belongs to the Indo-European language family. It is a morphologically complex language with very rich derivational and inflectional morphology. It has seven noun cases, a complex declension and conjugation system, pronominal clitics and other morpho-syntactic structures, which all make the Czech language difficult for language learners. The CzeSL corpus is unique because it provides opportunities for a researcher to analyze learners' linguistic output of a highly inflectional target language. The corpus consists of several sub-corpora, with a total of 2 million words.

Out of this, about 200K words are corrected and error annotated using a two level annotation scheme. The first layer corrects individual words disregarding their context, for example spelling errors. In addition to manually annotated tags, e.g., error in ending (*incorInfl*) or error in stem (*incorBase*), some tags are added automatically, e.g., missing vowel accent (*formQuant0*) or erroneous character substitution (*formSingCh*). The second layer describes corrections within context that concern mostly morpho-syntactic and stylistic errors, e.g., the valency error (*dep*) includes noun declension and verb-noun agreement errors. For our purpose, we use both layers of annotation. The tagset is described in the annotation manual (Štindlová et al., 2012), in addition to the papers mentioned above.

Each document in the corpus is labeled with metadata information, including the author's proficiency level and native language background. The essays are encoded in the Prague Markup Language format.<sup>1</sup>

---

<sup>1</sup> <http://ufal.mff.cuni.cz/jazz/pml/>

We report our findings for a binary classification between IE and NIE language backgrounds. We use 38<sup>2</sup> essays for lower intermediate (A2) and 38 essays for intermediate (B1) levels of proficiency. The essays are equally distributed among the language backgrounds within these levels. The essays are written on several topics, which are consistent throughout the groups. The topics include "My life in Prague", "The best/worst day in my life" and "Holidays", among several others. Every essay is written by a different author.

#### 3.2 Native Speakers' Predictions

Prior to implementing machine learning classification, we decided to conduct an experiment with native speakers of Czech using the same data. The motivation for this experiment is to see whether it is possible for a native speaker to predict the learners' IE vs. NIE language backgrounds based on their essays.

We asked 24 Czech native speakers with NLP and/or linguistic backgrounds to read the essays and make their predictions about the language background of the writers. To avoid content bias, we substituted all proper names of places with a capital X; and personal names with generic names across all essays, e.g., Eva and Pavel.

There were a total of 76 essays to evaluate. An online questionnaire was created,<sup>3</sup> where native speakers read as many randomly assigned essays as they wanted and filled in the keys according to their predictions. The possible answers were "IE", "NIE" and "unclear". As the result of this experiment, an average accuracy of 55% was achieved. This result is only slightly better than the baseline of 50% for two-group classification.

The participants of our experiment all had some training in linguistics. This suggests that if the participants did not have any linguistic background, their performance on the task would probably be even lower. Moreover, the essays could have still contained some contextual cues about the authors' background, which might have triggered a higher result as well. The partic-

---

<sup>2</sup> The CzeSL corpus is in final stages of development prior to public release. We used only those essays where we had access to all of: data itself, error annotation, morphological annotation and L1 metadata. We are planning to repeat the experiments, once there are more essays with such properties available.

<sup>3</sup> Using the open-source system developed by Jan Štěpánek <https://github.com/choroba/inquiry/>

ipants expressed their intuitions about the predictions. Specifically, they said they looked at the way the essays were written, the overall amount of errors. If an essay was written reasonably well the participants assumed that the author belonged to the IE group of learners, and vice versa. However, even having these intuitions in mind, our participants' performance was only slightly better than chance.

Overall, the experiment provided interesting observations and guided us towards a machine learning experiment.

### 3.3 SVM Classification

#### Data representation

For this experiment, our first goal is to see whether machine learning techniques are able to categorize the same set of data at higher performance rates than human native speakers using non-content based features. Our second goal is to see whether the native speakers' intuitions can be validated, specifically if it is the number of errors or other criteria that help to discriminate between the two classes.

We use the SVM-light classifier (Joachims, 1999). Each feature value is represented as a term weight of the feature, computed as a logarithmic ratio of the token frequency in the file to the total amount of tokens in the file.

$$S_{ij} = \text{round} (10 \times (1 + \log (tf_{ij})) / (1 + \log (l_j)))$$

Equation 1. The formula for computing the term weight of a feature where  $S_{ij}$  is a term weight,  $tf_{ij}$  is the number of occurrences of term  $i$  in document  $j$ , and  $l_j$  is the length of the document. (Manning and Schuetze, 1999, p.580).

The feature set includes 264 most frequent POS bi-grams (3 or more occurrences in the data), 305 most frequent POS tri-grams and 35 error types extracted from the corpus. The total of POS n-grams for all essays amounts to 20,000. For error types, the total amount of error type tokens is 2000. After preprocessing, each essay is characterized by a vector with no more than 604 dimensions.

We report the classification results for the best performing parameters ( $C, \gamma$ ) of the radial basis function (RBF) kernel SVM on the data set. Classification is performed by running the leave-one-out cross validation technique.

## Results

Our best model is trained on a corpus that contains essays of the lower intermediate level of learners and receives 89% precision and the same recall using only orthographic types of errors as features. This is almost 40% higher than the baseline of 50% for the two-class classification. The precision and recall measures for each experiment are described in Tables 1-3.

Features	Precision	Recall
POS bigrams	<b>78</b>	74
POS trigrams	70	74
Errors	70	<b>78</b>
Errors+POS n-grams	71	75

Table 1. Classifier performance on Level B1 (intermediate) Czech

Features	Precision	Recall
POS bigrams	70	74
POS trigrams	70	78
Errors	<b>89</b>	<b>89</b>
Errors+POS n-grams	78	<b>95</b>

Table 2. Classifier performance on Level A2 (lower intermediate) Czech

Features	Precision	Recall
POS bigrams	74	<b>89</b>
POS trigrams	68	79
Errors	84	84
Errors+POS n-grams	<b>85</b>	<b>89</b>

Table 3. Classifier performance on Level A2 + Level B1 (combined) Czech

The results also demonstrate that the error features of the two levels combined perform distinctively well, at 84%. All features together show 85% precision and 89% recall. From the above experiments, we can see that non-content features such as POS tags and error tags perform well for highly inflectional language data. Moreover, error tags, on their own, may be considered good indicators of a class for this classification. Using features that do not reflect content makes our method more general and topic- and genre independent.

### 3.4 Classification experiment using error tags only

Following the native speakers' intuitions from the experiment described in Section 3.2, we can assume that the discriminative power of errors should not be surprising; learners of Czech of a NIE language background are likely to make more errors than the learners of the IE group due to the differences between L1 and L2. However, we need to perform a more detailed error analysis to conform or disagree with these intuitions. The SVM classifier performs fairly well by using only error tags as features. In this section, we further investigate the results of the previous experiment, considering each feature separately. To verify what error types are good markers for the two groups, we run additional classification experiments on each error-tag feature using the Weka implementation of the Naïve Bayes classifier (Witten et al., 2011). Naïve Bayes is a probability-based classifier. It implements Bayes' Theorem, the basic idea of which is the independence assumption, i.e. the presence or absence of one feature does not depend on the presence or absence of another feature. Naïve Bayes is simple to implement and interpret. We perform the 10-fold cross-validation technique on each data set for this task. We report the precision and F-measure which are calculated from the feature values normalized by total token amounts.

#### Results

The results of the Naïve Bayes classification experiments for both levels of proficiency are described in Table 4 and Table 5. The best performing features are shown in bold.

Table 4 displays the results for the intermediate level (B1) with morpho-syntactic and stylistic errors, the second layer in CzeSL. At this level, 5 errors out of 13 perform with precision and F-measure higher than 50%. These errors are the errors in valency (*dep*), errors in incorrect use of bookish, dialectal expressions and hypocorrections (*stylOther*), misuse of grammatical forms (*use*), and odd constituent error (*odd*). The results suggest that these types of errors mostly contribute to the classification performance at this level.

Table 5 shows the results for the lower intermediate level (A2), the first layer of corrections in CzeSL. This level contains corrections of word-level errors, often of orthographic character.

The errors that perform with precision and F-measure higher than 50% (6 out of 22) are missing vowel accent (*formQuant0*), erroneous character substitution (*formSingCh*), incorrect use of 'i' instead of 'y' (*formY0*) ('i' and 'y' have the same pronunciation in Czech), incorrect use of 'y' instead of 'i' (*form Y1*), errors in inflections (*incorInfl*), and errors in stems (*incorBase*).

We also calculate error scores in order to identify which group (IE or NIE) tends to make more errors. The error scores are the ratios of the total frequency of an error type for all files to the total amount of errors in files.

The results of the Naïve Bayes classification suggest that depending on their nature, some errors contribute significantly to classification performance, but some have low discriminative power. For our purposes, these results are important for further analysis of the variety in the performance of the errors.

## 4 Discussion

Our results demonstrate that written texts regardless of the level of proficiency can be classified at 85% precision using non-content features, POS n-grams, and error tags combined. The results show that error-based features of two levels combined demonstrate a high performance of 84% suggesting that error annotated written learner Czech data provide reliable cues for distinguishing between the learners' IE and NIE backgrounds. The results show 89% precision and the same recall for a lower intermediate level (A2), which is annotated mostly for orthographic errors. The errors at the intermediate level (B1) with error tags of morpho-syntactic and stylistic character perform lower, at 70% precision and 78% recall.

The significant difference in the precision between the two levels suggests that the errors made by learners of a lower level of proficiency discriminate better than the errors made by higher level, i.e. if we have a fairly advanced learner, it would be harder to predict his or her language background. These results are not surprising, though more evidence is needed. It is more important to point out that the noticeably higher performance of orthographic errors suggests that these errors discriminate well between two language backgrounds within one level of proficiency. Consequently, this means that learners of two language backgrounds make errors specific for their L1 group. These results can be compared with previous observations made by other

researchers (Tsur et al., 2007; Kochmar, 2011) that character n-grams extracted from learner data contribute significantly to classification performance. As Tsur et al. (2007) hypothesized, the learners' choice of characters in L2 reflects the phonology of their L1. Although we do not specifically look at character n-grams, the better discriminating power of errors of orthographic nature achieved in our experiments might as well reflect spelling conventions of a specific L1 group. Thus, such errors are more likely to be learners' L1 to L2 transfer errors.

Below, we provide a more detailed error analysis to verify the nature of errors and possible reasons for their contribution to the performance.

#### 4.1 Error Analysis

Table 4 describes the results for Level B1. Lexical error (*lex*) shows 69% precision and 66% F-measure. These are lexicon or phraseology errors which occur, for instance, when learners misuse prepositions, choose false friends or false cognates instead of the correct variants. In the phrase *dopadlo to přírodně* the use of the adverb *přírodně* in this context results in an error. The intention of the author is more likely to say 'it ended naturally', but the word *přírodně* means 'naturally' in a sense of nature/non-artificial. The error scores show that learners of the IE language background tend to make more errors of this type (19.9/11.5). IE learners might use false cognates in Czech more often because of the similarity between L1 and L2 languages, e.g., Russian and Polish.

Stylistic errors (*stylOther*) reflect stylistic discrepancies, such as misuse of bookish, dialectal forms, slang, and hyper-corrections. For instance, in the phrase *pláči nad vejdělkem* 'be unhappy about the result' the correct use will be *pláču nad vydelkem*. There is a hypercorrection in the use of *pláči* instead of *pláču*. These types of errors occur exclusively within the IE group of learners (5.4/0) and perform with the highest precision of 79% and F-measure of 59%. This result together with the observations made for the *lex* type of error suggests that the L1s which are closer related to Czech influence the production of the L2 in a less subtle way than more distant languages. Specifically, the use of cognates in the incorrect context or use of incorrect stylistic variants might result in a transfer error in this case (Kroll et al., 2002).

The valency error (*dep*), e.g., using *bojí se pes* instead of *bojí se psa* 'he is scared of a dog' (dog

is a direct object, thus accusative *psa* instead of nominative *pes* must be used) yields 61%

Error Type	Precision	F-measure
agr	46	42
dep	<b>61</b>	<b>56</b>
lex	<b>69</b>	<b>66</b>
miss	50	49
stylOther	<b>79</b>	<b>59</b>
use	<b>64</b>	<b>64</b>
odd	53	51
sec	50	49
rflx	19	19
stylCol	50	44
vbx	46	44
cvf	44	39
ref	50	46

Table 4. Results of Naïve Bayes classification, Level B1.

Error Type	Precision	F-measure
formCap1	50	41
formCaron0	43	42
formCaron1	46	42
formQuant0	<b>76</b>	<b>73</b>
formReduChar	56	45
formSingCh	<b>74</b>	<b>70</b>
formVoiced	40	37
formY0	<b>70</b>	<b>55</b>
formY1	<b>81</b>	<b>65</b>
incorBase	<b>80</b>	<b>79</b>
incoInfl	<b>67</b>	<b>65</b>
styCol	39	39
wbdOtherJnt	77	49
flex	64	47
formQuant1	50	47
formVoiced0	50	38
fwNc	36	35
fwFab	50	43
missChar	60	49
wbdPreSplit	41	36
formDiaE	41	36
formMeta	76	44

Table 5. Results of Naïve Bayes classification, Level A2.

precision and 56% F-measure. Valency errors reflect the differences between the morpho-syntactic structures of L1 and L2. The use of grammar category (*use*) type of error shows 64% precision and 64% F-measure. This type includes errors in tense and aspect, incorrectly

formed comparative, and singular instead of plural among others. For instance, using *včera bude sněžit* ‘yesterday it will snow’ instead of *včera sněžilo* results in a verb tense error, future form *bude sněžit* is used instead of the past *sněžilo*. The two errors above largely concern Czech morpho-syntax. Thus, greater differences between L1 and L2 grammatical structures might trigger a higher amount of errors within the NIE group. However, a more detailed analysis of error distribution within each group and probably a larger data set are needed to investigate this claim.

The results for Level A2 are displayed in Table 5. The missing vowel accent (*formQuant0*) error occurs in such cases as incorrect *vzpomínám* vs. correct *vzpomínám*, or *doufam* vs. *doufám*. This error type performs at 76% precision and 73% F-score. The erroneous character substitution error (*formSingCh*), e.g., incorrect *otevrila* vs. correct *otevrela* or *vezmíme* vs. *vezmeme*, performs well at 74% precision and 70% F-measure. The error scores show that the NIE learners tend to make more errors of this type (6/11). Errors in inflectional endings (*incolInfl*), e.g., using *plavám* instead of *plavu* ‘I swim’ (1Sg ending ‘-ám’ of one paradigm is used with a verb of another), perform at 67% precision and 65% F-measure. Errors in stems (*incorBase*) e.g., using *dítem* instead of *dítětem*, discriminate rather well, at 80% precision and 79% F-measure. The two types of errors that describe incorrect use of ‘i’ and ‘y’ (*formY0* and *formY1*, respectively) show high precision (70% and 81%) and F-measure (55% and 65%). The NIE learners make more errors of type *Y0* (1.4/7.4), e.g., *pražských* instead of *pražských*, *vypije* vs. *vypije*, whereas the IE group solely makes errors in the other type, *Y1* (1.5/0), e.g., *hlavným* instead of *hlavním*, *líbyl* vs. *líbil*. The above results suggest that learners might make some motivated spelling choices related to their L1 backgrounds (Jarvis et al., 2012; Tsur et al., 2007). For instance, speakers of Russian and Belarusian, IE group, would use the letter ‘y’ more often in the ending because it corresponds to the phonological equivalent of the letter ‘ы’ of the Cyrillic alphabet, e.g., *главным* in Russian will be a phonological equivalent of incorrect *hlavným* in Czech.

Our analysis of the best performing features shows that learners of *both* language backgrounds within each level of proficiency produce errors that discriminate well and vary in nature between IE and NIE learners. Thus, we cannot

strictly follow the intuition that the NIE learners make more errors, although these results might change with a larger number of learner essays. The error analysis at Level B1 shows that the best performing morpho-syntactic errors occur more within the NIE group of learners, whereas errors of stylistic and lexical character discriminate better for the IE group compared to NIE within the same level of proficiency. From the analysis of Level A2, we can conclude that the learners’ L1 can be traced from some errors which provides evidence for L1 to L2 transfer. At the same time, for other errors, it is impossible to identify their nature and group prevalence based on the data available.

Our results also suggest that a wide range of manually and automatically annotated error tags is a valuable venue to explore in the context of native language identification. Error-based features have been approached by other researchers as it is mentioned in Section 2. Koppel et al. (2005) use 185 error types, which do not appear to contribute significantly to the performance. Wong and Dras (2009) use only three features which do not improve the overall results, and do not perform as high as our error-based features on their own. Kochmar (2011) provides a very systematic error analysis and conducts a number of two-group classifications. Her results demonstrate that using character quad-grams achieves the highest precision of 100% for the Danish - Swedish group. However, the author points out that character quad-grams are likely to create content bias. In our case, tags are used for errors and a high result is achieved.

At the same time, our results cannot be directly compared with the studies described above for several reasons. First, we formulate our task differently – we only identify the learners’ L1 language family rather than a specific language. Second, we use a language with a different and more complex morphological structure which might have caused a large amount of learner errors and thus, provided with the discriminative power of these features. Third, we use essays of an intermediate level of proficiency which might have contained more errors than the intermediate to advanced levels discussed previously (Argamon et al., 2009; Tsur et al., 2007). As we emphasized above, we intentionally do not use lexical features such as function words, because some function words might reflect the content of the essays to a higher degree than other, e.g., pronouns or prepositions. For instance, if learners write about their daily routines they tend

to use more prepositions and adverbs of time. If learners write about themselves, they tend to use personal pronouns. Argamon et al. (2009) describe function words as the most effective features for the task. However, when interpreting their results, we should keep in mind that the results might be artifacts of topical bias, since the topics were not strictly controlled in this study.

## 4.2 SLA Implications

We believe that our results are important for SLA. In particular, the results provide empirical evidence for the different types of errors discussed within the Error Analysis approach. We observe that some of the best performing errors occur when a learner's L1 interferes and affects the production of L2. We suggest that some of the highly discriminative spelling errors at the lower intermediate level are likely to be transfer errors for both groups, in support of the observations made by other researchers in regards to character n-grams. We also suggest that some stylistic errors are highly discriminative at the intermediate level within the IE group. At the same time, some of the errors that occur often within both IE and NIE groups might be developmental, and at this point these observations are not completely evident. Further experiments with more fine-grained error annotation and linguistic analysis might provide better insights on whether the best performing errors are of interlingual or developmental character. Overall, our results suggest that native speakers of Indo-European and non-Indo-European languages approach Czech differently, in their specific L1 background ways and make consistent types of errors across different linguistic levels, in particular lexicon and orthography, based on our data.

## 5 Conclusions and Future Work

We have described several experiments in which we explore various features to distinguish between two large language groups of learners, Indo-European and non-Indo-European. We have addressed non-content based features, and have shown that they work well for highly inflectional data. Exploring non-content based features is important because it provides opportunities for corpus independent approaches for native language identification. We have also discussed the best performing features and their contribution to the task.

Section 4.2 discussed the implications of our work to the field of SLA, but this work has further applications. By knowing what typical errors L1 learners make, language instructors can concentrate on helping their students to erase their “non-native” footprints. Other applications include marketing research, automatic error-correction and grading applications.

Our results go along with similar observations made for learner English data, that data-driven machine learning approaches are valuable for verifying SLA hypotheses (Jarvis et al., 2012). In addition, we look at Czech as the target language, which has not been discussed in the context of language background identification thus far, to the best of our knowledge. Also, our data shed light on the acquisition of target languages with complex morphology.

As for the future directions of our work, we would like to develop methods to derive best performing error tags automatically. Further, we would like to perform experiments with larger sets of data and to compare the performance of features for other levels of proficiency. Ultimately, we would like to develop a method that will be able to make more fine-grained distinctions between learners' language backgrounds using non-content based features and pin down the actual native language of the learner based on this type of data.

## Acknowledgements

We would like to thank the native speakers of Czech for their participation in our experiment and to Jan Štěpánek for tailoring his questionnaire system to our needs. We would also like to thank Jing Peng and the anonymous reviewers for their comments.

This material is based in part upon work supported by the Grant Agency of the Czech Republic P406/10/P328 and National Science Foundation under Grant Numbers 0916280 and 1048406. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

- Shlomo Argamon, Moshe Koppel, James W. Pennebaker, and Jonathan Schler. 2009. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52: 119-123.



- Stephen P. Corder. 1975. Error analysis, interlanguage and second language acquisition. *Language Teaching*, 8:201-218.
- Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford. 2007. TAT: An author profiling tool with application to Arabic emails. In *Proceedings of the 5<sup>th</sup> Australasian Language Technology Workshop*, pages 21-30.
- Sylviane Granger. 2003. The International Corpus of Learner English: A new resource for foreign language learning and teaching second language acquisition research. *TESOL Quarterly*, 37:538-546
- Jirka Hana, Alexandr Rosen, Svatava Škodová, and Barbora Štindlová. 2010. Error-tagged Learner Corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV)*, Uppsala, pages 11-20.
- Hans van Halteren. 2004. Linguistic profiling for author recognition and verification. In *Proceedings of the 42<sup>nd</sup> Annual Meeting on Association for Computational Linguistics (ACL'04)*, page 199-207.
- Scott Jarvis and Scott A. Crossley, editors. 2012. *Approaching Language Transfer through Text Classification: Explorations in the Detection-based Approach*. Multilingual Matters, UK, US, Canada.
- Tomáš Jelínek, Barbora Štindlová, Alexandr Rosen, and Jirka Hana. 2012. Combining manual and automatic annotation of a learner corpus. *Text, Speech and Dialogue Lecture Notes in Computer Science*, 7499:127-134.
- Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML - 98*, pages 137-142.
- Thorsten Joachims. 1999. Making large-scale SVM learning practical. In Bernhard Schoelkopf and Christopher Burges and Alexander Smola, editors. *Advances in Kernel Methods – Support Vector Learning*, pages 169-185. MIT Press.
- Ekaterina Kochmar. 2011. *Identification of a writer's native language by error analysis*. Master of Philosophy Thesis. University of Cambridge, 2011.
- Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, pages 624-628.
- Judith F. Kroll, Erica Michael, Natasha Tokowicz, and Robert Dufour. 2002. The development of lexical fluency in a second language. *Second Language Research*, 18:137-171.
- Robert Lado. 1957. *Linguistics across Cultures: Applied Linguistics for Language Teachers*. University of Michigan Press, Ann Arbor, MI, US.
- Christopher D. Manning and Hinrich Schuetze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, US.
- F. Mosteller and D.L. Wallace. 1984. *Applied Bayesian and Classical Inference in the Case of the Federalist Papers* (2<sup>nd</sup> edition). Springer Verlag, New York.
- Jack C. Richards. 1971. A non-contrastive approach to error analysis. *ELT Journal*, 25:204-219.
- Alexandr Rosen, Jirka Hana, Barbora Štindlová, and Anna Feldman. 2013. Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaluation*, 47:1-28.
- Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. 2001. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35: 193-214.
- Štindlová Barbora and Alexandr Rosen. 2012. Návod k anotaci chybového korpusu [Learner Corpus Annotation Manual]. Unpublished. <http://utkl.ff.cuni.cz/~rosen/public/anotace.pdf>
- Oren Tsur and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9-16.
- Ian H. Witten, Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Tools and Techniques*. 3d Edition, Morgan Kaufmann, San Francisco, 2011.
- Sze-Meng Jojo Wong and Mark Dras. 2009. Contrastive analysis and native language identification. In *Proceedings of the Australian Language Technology Association Workshop*, pages 53-61.

# Author Index

- A, Ambha, 733  
Abou El-Nasr, Mohamad, 471  
Afli, Haithem, 286  
Agarwal, Apoorv, 1202  
Agrawal, Bhasha, 1237  
Aha, David, 747  
Aharodnik, Katsiaryna, 1428  
Aizawa, Akiko, 809, 1383  
Akbik, Alan, 1312  
Akula, Arjun, 1216  
Al-Sabbagh, Rania, 410  
Ali, Tanveer, 667  
Alotaibi, Fahd, 392  
Anai, Hirokazu, 73  
Andrade, Daniel, 127, 1077  
Aono, Masaki, 932  
Arai, Noriko, 73  
Araki, Kenji, 579  
Aramaki, Eiji, 874  
Atreya V, Arjun, 982
- Baek, Seung-Cheol, 699  
Bai, Ming-Hong, 839  
Bairi, Ramakrishna, 733  
Baldwin, Timothy, 356, 685  
Ballesteros, Miguel, 942  
Bandyopadhyay, Sivaji, 674, 892  
Banerjee, Somnath, 892  
Bangalore, Srinivas, 1032  
Barrault, Loïc, 286  
Behera, Bibek, 937  
Beinborn, Lisa, 883  
Belguith, Lamia Hadrich, 419  
Berka, Tobias, 1002  
Bhat, Riyaz Ahmad, 64, 189  
Bhattacharyya, Pushpak, 661, 774, 937, 982  
Biran, Or, 788  
Bird, Steven, 1134, 1243  
Bohnet, Bernd, 942, 1250  
Bouamor, Dhouha, 952  
Bouamor, Houda, 270  
Boudin, Florian, 543, 834  
Bougouin, Adrien, 543  
Boujelbane, Rahma, 419
- Brooke, Julian, 82  
Bunescu, Razvan, 498  
Burga, Alicia, 1250
- Calvanese, Diego, 1129  
Can, Burcu, 1087  
Cano Basave, Amparo Elizabeth, 109  
Cao, Yunbo, 28  
Cardillo, Elena, 1129  
Carrol, John, 1012  
Catherine, Rose, 1  
Cattle, Andrew, 864  
Chali, Yllias, 767  
Chang, Baobao, 1271  
Chang, Chin-Ting, 802  
Chang, Chuang-Ping, 561  
Chang, Jason S., 706, 839  
Chang, Marco, 1428  
Chao, Lidia S., 907  
Chen, Chen, 822, 1366  
Chen, Huan, 320  
Chen, Keh-Jiann, 839  
Chen, Kuan-Yu, 1117  
Chen, MeiHua, 706  
Chen, Qingcai, 726  
Chen, Tongfei, 1278  
Chen, Xiaoxin, 1209  
Chen, YiChun, 706  
Chen, Yun-Nung, 648  
Chen, Zhenbiao, 972  
Cheng, Xueqi, 1139  
Cho, Heeryon, 463  
Chu, Chenhui, 1144  
Chu, Cuong, 740  
Cicekli, Ilyas, 854, 1230  
Clifton, Ann, 1072  
Cook, Paul, 356, 1243
- Daille, Béatrice, 401, 543  
Dakwale, Praveen, 977  
Dandala, Bharath, 498  
Das, Dipankar, 674  
Dey, Kuntal, 1375  
Diab, Mona, 1181

Dickinson, Markus, 1419  
Diesner, Jana, 410  
Ding, Chenchen, 516  
Ding, Xiao, 311  
Ding, Zhuoye, 118  
Dong, Xishuang, 455  
Duh, Kevin, 781  
Duong, Long, 1243  
  
Ebrahimi, Sajad, 1151  
Eccher, Claudio, 1129  
Eiselt, Andreas, 829  
Ekbal, Asif, 815  
El Kholy, Ahmed, 1047, 1174  
El-Sonbaty, Yasser, 471  
Ellouze Khemakhem, Mariem, 992  
Ellouze Khemekh, Mariem, 419  
Engels, Gregor, 534  
  
Farkas, Richárd, 329  
Feldman, Anna, 1428  
Feng, Vanessa Wei, 338  
Feng, Xiao, 859  
Fennell, Jason, 922  
Figuerola, Alejandro, 829, 902  
Foster, Jennifer, 1092, 1167  
Fu, Lisheng, 692  
Fu, Xiaoyin, 972  
Fujii, Atsushi, 878  
  
Gandhe, Ankur, 429  
Gangadharaiah, Rashmi, 1, 243, 429  
Gao, Wenliang, 1107  
Ghoneim, Mahmoud, 1181  
Girju, Roxana, 410  
Goldenberg, Benjamin, 922  
Gong, Yeyun, 118  
Grishman, Ralph, 692  
Guan, Yi, 455  
Gurevych, Iryna, 883, 1188  
  
Habash, Nizar, 1047, 1174  
Hadrach Belguith, Lamia, 992  
Haffari, Gholamreza, 438  
Hajičová, Eva, 55, 91  
Hana, Jirka, 1428  
Hanke, Florian R., 1134  
Harastani, Rima, 401  
Hasan, Kazi Saidul, 1348  
Hasan, Sadid A., 767  
Hazem, Amir, 1392  
He, Liangye, 907  
He, Yulan, 109  
  
Hirst, Graeme, 82, 338  
Hoenen, Armin, 1299  
Hollowood, Fred, 1092, 1167  
Hoshino, Sho, 1062  
Hou, Wen-Juan, 561  
Hsieh, Yu-Ming, 839  
Hu, Junfeng, 1278  
Hu, Yuxiu, 1067  
Huang, Chien-Kang, 614  
Huang, Chu-Ren, 795  
Huang, Chung-chi, 912  
Huang, Lian'en, 10  
Huang, ShihTing, 706  
Huang, Xuanjing, 118, 320, 552  
  
Imamura, Kenji, 1292  
Inkpen, Diana, 667  
Inui, Kentaro, 587  
Ishida, Toru, 1052  
Ishikawa, Kai, 127, 1077  
Islam, Zahurul, 1299  
Israel, Ross, 1419  
Iwane, Hidenao, 73  
  
Jain, Naman, 189  
Jain, Sambhav, 189, 1082, 1237  
Jauhar, Sujay Kumar, 648  
Jeong, Yoonjae, 136  
Jia, Zhongye, 1195  
Jiang, Han, 507  
Jin, Gongye, 947  
Jing, How, 1117  
Jínová, Pavlína, 91  
Johannsen, Anders, 987, 997  
John, Ajita, 365  
Jönsson, Arne, 1223  
Joshi, Sachindra, 570  
  
Kaji, Hiroyuki, 1057  
Kaji, Nobuhiro, 153, 1107  
Kakde, Yogesh, 982  
Kan, Min-Yen, 127  
Kando, Noriko, 917  
Kawahara, Daisuke, 37, 171, 947  
Kawahara, Tatsuya, 957  
Kawazoe, Ai, 1357  
Khadivi, Shahram, 1151  
Khaliq, Bilal, 1012  
Kim, Jin-Dong, 1112  
Kim, Munhyong, 864  
Kim, Songkuk, 463  
Kim, Su Nam, 225, 234

Kim, Youngsam, 145, 864  
Kimura, Yasutomo, 579  
Kirschnick, Johannes, 1312  
Kitsuregawa, Masaru, 153, 1107  
Klakow, Dietrich, 19  
Kleinbauer, Thomas, 225, 234  
Koike, Daichi, 917  
Kosseim, Leila, 1401  
Kotalwar, Anup, 1202  
Kristianto, Giovanni Yoko, 809  
Ku, Lun-Wei, 912  
Kurohashi, Sadao, 37, 162, 171, 261, 947, 1144  
Kutlu, Mucahid, 1230

Lai, Siwei, 1097  
Lang, R. Raymond, 1306  
Lapata, Mirella, 507  
Lau, Jey Han, 685  
Le Roux, Jonathan, 962  
Lee, Chia-ming, 614  
Lee, Jong-Seok, 463  
Lee, Mark, 392  
Lee, Sun-Hee, 1419  
Leusch, Gregor, 1174  
Li, Sheng, 596, 967  
Li, Wenjie, 641  
Li, Xiaoming, 507  
Liang, Wei, 859  
Lin, Chin-Yew, 28  
Lin, Donghui, 1052  
Lin, Shou-De, 507  
Litvak, Marina, 655  
Liu, Bingyang, 1139  
Liu, Chengyong, 859  
Liu, Fang, 1097  
Liu, Jiangming, 927  
Liu, Kang, 109, 1097  
Liu, Lema, 279  
Liu, Qian, 1139  
Liu, Qun, 447  
Liu, Ting, 311, 480  
Liu, Tse, 802  
Liu, Yang, 1097  
Liu, Yue, 1139  
Liu, Zhiguang, 455  
Löser, Alexander, 1312  
Lu, Bao-Liang, 605  
Lu, Shixiang, 972  
Lui, Marco, 356  
Luo, Yanyan, 781  
Lv, Xueqiang, 507  
Lv, Yajuan, 447

MacKinlay, Andrew, 356  
Mai Xuan, Trang, 1052  
Majumder, Prasenjit, 1102  
Malladi, Deepak Kumar, 1007  
Mamidi, Radhika, 1216  
Manandhar, Suresh, 1087  
Mannem, Prashanth, 1007  
Manning, Christopher D., 525, 1285  
Mansur, Mairgup, 1271  
Martínez, Héctor, 942  
Martínez-Gómez, Pascual, 1383  
Maskawa, Sachiko, 874  
Maskharashvili, Aleksandre, 1257  
Masui, Fumito, 579  
Matsumoto, Yuji, 781  
Matsuzaki, Takuya, 73  
Matusov, Evgeny, 1174  
McKeown, Kathleen, 788, 1124, 1410  
Mehta, Parth, 1102  
Mehta, Sameep, 1375  
Meng, Fanqi, 641  
Meng, Yao, 869  
Merhbene, Laroussi, 1027  
Mersch, John, 1306  
Meshgi, Kourosh, 1151  
Metze, Florian, 648  
Mihalcea, Rada, 498  
Mírovský, Jiří, 55, 91  
Misra Sharma, Dipti, 1082  
Mistica, Meladel, 685  
Mithun, Shamima, 1401  
Miyabe, Mai, 874  
Miyao, Yusuke, 753, 1062, 1357  
Mizuno, Junta, 587  
Mohit, Behrang, 270  
Mollá-Aliod, Diego, 712  
Montali, Marco, 1129  
Mori, Shinsuke, 957  
Morin, Emmanuel, 401, 1392  
Morita, Mizuki, 874  
Moschitti, Alessandro, 100, 1330  
Mujadia, Vandan, 977  
Mukherjee, Subhabrata, 570  
Mulholland, Matthew, 680  
Murakami, Yohei, 1052  
Murawaki, Yugo, 46  
Myaeng, Sung-Hyon, 136

N, Vasudevan, 774  
Nagar, Seema, 1375  
Nagata, Masaaki, 1062  
Nagy T., István, 207, 329

Nakayama, Yuki, 878  
Nakazawa, Toshiaki, 261, 1144  
Narang, Kanika, 1375  
Narayanaswamy, Balakrishnan, 243  
Narita, Kazuya, 587  
Nedoluzhko, Anna, 91, 1037  
Neumann, Guenter, 902  
Neviarouskaya, Alena, 932  
Ng, Vincent, 822, 1348, 1366  
Nghiem, Minh-Quoc, 809  
Nguyen, Dai Quoc, 897  
Nguyen, Dat Quoc, 897  
Nguyen, M.L., 1042  
Nguyen, Minh, 740  
Nguyen, Minh Le, 1017  
Nguyen, Ngan, 753  
Nguyen, Phuong-Thai, 1042  
Nguyen, Tung-Lam, 1042  
Nitta, Taisei, 579  
Novák, Michal, 1037

Oflazer, Kemal, 270  
Okumura, Manabu, 162, 674  
Oliveira, Francisco, 907  
Onishi, Takashi, 127, 1077  
Oraby, Shereen, 471  
Otmakhova, Julia, 864

Padró, Muntsa, 942  
Pala Er, Nagehan, 854  
Palshikar, Girish, 1264  
Paris, Cécile, 712  
Park, Jong, 699  
Park, Suzi, 864  
Patil, Sangameshwar, 1264  
Patra, Braja Gopal, 674  
Patwardhan, Siddharth, 1330  
Pawar, Sachin, 1264  
Pecina, Pavel, 1243  
Pei, Wenzhe, 1271  
Peng, Bo, 10  
Peng, Xiaochang, 180  
Petinot, Yves, 1124  
Pham, Minh Quang Nhat, 1017  
Pham, Son, 740  
Pham, Son Bao, 897  
Poesio, Massimo, 815  
Pogodalla, Sylvain, 1257  
Poláková, Lucie, 91  
Pollock, Colin, 922  
Popescu, Octavian, 347  
Prabhakaran, Vinodkumar, 216, 365

Ptaszynski, Michal, 579

Qian, Jin, 320  
Qin, Bing, 311, 480  
Qin, Yang, 1067  
Qin, Yanxia, 302  
Quinn, Joanne, 680

R. Hershey, John, 962  
Raghu, Dinesh, 1  
Ramakrishnan, Ganesh, 733, 982  
Rambow, Owen, 216, 1202  
Rangarajan Sridhar, Vivek Kumar, 1032  
Rasooli, Mohammad Sadegh, 1047  
Razmara, Majid, 252  
Richardson, John, 261  
Roturier, Johann, 1092, 1167  
Rubino, Raphael, 1092, 1167  
Rysová, Kateřina, 55  
Rysová, Magdaléna, 55  
Rzepka, Rafal, 579

Saadane, Houda, 1022  
Saers, Markus, 1158  
Saggion, Horacio, 374  
Saha, Sriparna, 815  
Samad Zadeh Kaljahi, Rasoul, 1092, 1167  
Sangal, Rajeev, 1216  
Sankaran, Baskaran, 438, 1032  
Sarkar, Anoop, 252, 438, 1072  
Sarker, Abeed, 712  
Sasano, Ryohei, 162  
Sawaf, Hassan, 1174  
Scheible, Silke, 489  
Schnabel, Tobias, 198  
Schramm, David, 667  
Schulte im Walde, Sabine, 489, 632  
Schütze, Hinrich, 198, 844, 1321  
Schwenk, Holger, 286  
Sekine, Satoshi, 1339  
Seligmann, Dorée D., 365  
Semmar, Nasredine, 952, 1022  
Søgaard, Anders, 987, 997  
Sharma, Dipti, 189  
Sharma, Dipti M, 977  
Sharma, Dipti Misra, 64  
Sharma, Raksha, 661  
Shen, Mo, 171  
Shen, Yang, 859  
Shibata, Tomohide, 37  
Shimazu, Akira, 1017  
Shin, Hyopil, 145, 864

Shinzato, Keiji, 37, 1339  
Shiri Ahmad Abady, Mohammad Ebrahim, 1151  
Sikdar, Utpal, 815  
Silvervarg, Annika, 1223  
Sokolova, Marina, 667  
Song, Yan, 623  
Soni, Ankush, 1082  
Springorum, Sylvia, 489, 632  
Štajner, Sanja, 374  
Stein, Benno, 534  
Strapparava, Carlo, 347  
Su, Jian, 28  
Su, Jinsong, 447  
Subramaniam, L V, 1375  
Sudoh, Katsuhito, 1062  
Sumita, Eiichiro, 279  
Sun, Ke, 596  
Sun, Shuqi, 596  
Sun, Weiwei, 180  
Sun, Xu, 641  
Szarvas, György, 1188  
  
Takahashi, Yusuke, 917  
Takamura, Hiroya, 674  
Tammewar, Aniruddha, 189  
Tan, Chew-Lim, 28  
Thadani, Kapil, 1124, 1410  
Thamrongrattanarit, Attapol, 922  
Thorne, Camilo, 1129  
Tian, Liang, 907  
Topic, Goran, 809  
Tran, Dang, 740  
Tsai, Ming-Feng, 802  
Tsao, Yu, 1117  
Tsuchida, Masaaki, 1077  
Tsunakawa, Takashi, 1057  
  
Uryupina, Olga, 100, 815  
Uthus, David, 747  
Utsuro, Takehito, 917  
Utt, Jason, 632  
  
Vajteršic, Marian, 1002  
van der Plas, Lonneke, 844, 1321  
Vanetik, Natalia, 655  
Vincze, Veronika, 207, 329, 383  
Visengeriyeva, Larysa, 1312  
Visweswariah, Karthik, 1  
Vu Huy, Hien, 1042  
  
Wachsmuth, Henning, 534  
Wan, Xiaojun, 180  
Wang, Aobo, 127  
Wang, Chuan-Ju, 802  
Wang, Dandan, 726  
Wang, Haifeng, 293, 596  
Wang, Houfeng, 641  
Wang, Hsin-Min, 1117  
Wang, Li, 356  
Wang, Mengqiu, 525, 1285  
Wang, Xiaolin, 605  
Wang, Xiaolong, 726, 1067  
Wanner, Leo, 1250  
Watanabe, Shinji, 962  
Watanabe, Taro, 279  
Wei, Chongqiang, 1067  
Wei, Wei, 972  
Welty, Chris, 1330  
Whitney, Max, 1072  
Wiegand, Michael, 19  
Wilson, Shomir, 760  
Wong, Derek F., 907  
Wu, Dayong, 1139  
Wu, Dekai, 1158  
Wu, Jianwei, 869  
Wu, Xianchao, 849, 1209  
  
Xia, Fei, 623  
Xiang, Chao, 480  
Xiang, Yang, 1067  
Xiao, Rixin, 1209  
Xu, Bo, 972  
Xu, Hongzhi, 795  
Xu, Jinan, 927  
Xu, Juan, 552  
Xue, Han, 480  
  
Yamamoto, Mikio, 516  
Yamamoto, Yosuke, 1057  
Yan, Rui, 507  
Yang, Jinfeng, 455  
Yang, Muyun, 596, 967  
Yarmohammadi, Mahsa, 1032  
Yasuda, Sachi, 874  
Yin, Jie, 719  
Yoshinaga, Naoki, 1107  
Yoshino, Koichiro, 957, 962  
Yoshioka, Masaharu, 917  
Yu, Bingyang, 726  
Yu, Hao, 869  
Yu, Heng, 447  
  
Zabokrtsky, Zdenek, 1037  
Zavareh, Farshid, 234  
Zesch, Torsten, 883

Zhang, Chao, 293  
Zhang, Fan, 10  
Zhang, Min, 302  
Zhang, Qi, 118, 320, 552  
Zhang, Shu, 869  
Zhang, Shuwu, 859  
Zhang, Wei, 28  
Zhang, Yaoyun, 1067  
Zhang, Yue, 302  
Zhang, Yujie, 927  
Zhao, Hai, 605, 1195  
Zhao, Jun, 109, 1097  
Zhao, Shiqi, 293, 596  
Zhao, Tiejun, 279, 967  
Zheng, Dequan, 302, 869  
Zheng, Wen, 1067  
Zhou, Guangyou, 1097  
Zhou, Xiaoqiang, 1067  
Zhou, Yaqian, 118  
Zhu, Junguo, 967  
Zhu, Weimeng, 1278  
Ziering, Patrick, 844, 1321  
Zikánová, Šárka, 91  
Zou, Xiaojun, 1278  
Zouaghi, Anis, 1027  
Zribi, Inès, 992  
Zrigui, Mounir, 1027  
Zsibrita, János, 207  
Zukerman, Ingrid, 225, 234  
Zweigenbaum, Pierre, 952